

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

МАТЕМАТИЧНИЙ ФАКУЛЬТЕТ

Кафедра комп'ютерних наук

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

на тему: «**РОЗРОБКА ІНТЕРАКТИВНОГО
НАВЧАЛЬНОГО КУРСУ "ІНФОРМАТИКА–
7КЛАСС"»**

Виконала: студентка 2 курсу, групи 8.1228
спеціальності 122 комп'ютерні науки
освітньої програми комп'ютерні науки
(шифр і назва спеціальності)

Г.П. Ільїна

(ініціали та прізвище)

Керівник

доцент кафедри комп'ютерних наук,
доцент, к.п.н., Пшенична О.С.

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Рецензент

завідувач кафедри фізики та методики її
викладання, д. пед. н., Іваницький О.І.

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**

Факультет математичний

Кафедра комп'ютерних наук

Рівень вищої освіти магістр

Напрямок підготовки 122 комп'ютерні науки

(шифр і назва)

Освітня програма комп'ютерні науки

ЗАТВЕРДЖУЮ

Завідувач кафедри
комп'ютерних наук, к.т.н.,
доцент

Борю С. Ю.

(підпис)

« 29 » травня 2019 р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТЦІ

Ільїній Ганні Петрівні

(прізвище, ім'я та по-батькові)

1. Тема роботи (проекту) Розробка інтерактивного навчального курсу "Інформатика–7клас"
керівник роботи (проекту) Пшенична Олена Станіславівна, к.п.н., доцент
(прізвище, ім'я та по-батькові, науковий ступінь, вчене звання)

затверджені наказом ЗНУ від « 29 » Травня 2019 року № 811–с

2. Строк подання студентом роботи 25.12.2019

3. Вихідні дані до роботи 1. Постановка задачі.
2. Перелік літератури.

4. Зміст розрахунково–пояснювальної записки (перелік питань, які потрібно розробити)
1. Постановка задачі.
2. Основні теоретичні відомості.
3. Розроблений програмний продукт відповідно до теми кваліфікаційної роботи

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)
Презентація

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання 02.09.2019**КАЛЕНДАРНИЙ ПЛАН**

№	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Розробка плану роботи.	05.09.2019	
2.	Збір вихідних даних.	07.09.2019	
3.	Обробка методичних та теоретичних джерел.	15.09.2019	
4.	Розробка першого та другого розділу.	20.10.2019	
5.	Розробка третього розділу.	09.11.2019	
6.	Оформлення та нормоконтроль кваліфікаційної роботи.	27.12.2019	
7.	Захист кваліфікаційної роботи.	14.01.2020	

Студент _____
(підпис)Г.П. Ільїна
(ініціали та прізвище)Керівник роботи _____
(підпис)О.С. Пшенична
(ініціали та прізвище)**Нормоконтроль пройдено**
Нормоконтролер _____
(підпис)О.Г. Спиця
(ініціали та прізвище)

РЕФЕРАТ

Кваліфікаційна робота магістра «Розробка інтерактивного навчального курсу "Інформатика – 7 клас"»: 94 с., 27 рис., 13 табл., 11 джерел, 2 додатки.

АЛГОРИТМ, БАЗА ДАНИХ, ІНФОРМАТИКА, МОВА ПРОГРАМУВАННЯ, НАВЧАЛЬНИЙ КУРС, ПРОГРАМА, САЙТ, СКРИПТ, СИСТЕМА КЕРУВАННЯ ЗМІСТОМ, СТРУКТУРНА СХЕМА,.

Предмет дослідження – визначення принципів побудови інтерактивних навчальних курсів на основі web–ресурсу на базі CMS Drupal та PHP.

Об’єкт дослідження – інтерактивний курс навчання школярів інформатики на основі web технологій.

Мета роботи: розробити інтерактивний навчальний курс навчання інформатиці школярів 7 класу на базі web технологій.

Метод дослідження – оглядовий, аналітичний.

Для реалізації поставленої мети було вирішено такі завдання:

– на основі аналізу процесу навчання інформатики в школі запропоновано моделі цього процесу і структуру інтерактивної навчальної системи;

– розроблено методи і алгоритми з ймовірнісної оцінкою кількісних і якісних параметрів навчальної системи;

– обґрунтовано вибір застосування сучасних засобів для створення інтерактивних навчальних курсів з технологію їх використання;

– перевірено ефективність розробленої методики шляхом створення і експлуатації навчального курсу на базі інтегрованих програмних засобів.

При виконанні роботи застосовані система управління базами даних (СУБД) MySQL, система керування вмістом Drupal та скриптова мова програмування PHP.

SUMMARY

Master's Qualification Thesis "Development of an Interactive Training Course "Informatics – 7 class"": 94 pages, 27 figures, 13 tables, 11 sources, 2 applications.

ALGORITHM, DATABASE, INFORMATION, PROGRAMMING LANGUAGE, EDUCATIONAL COURSE, PROGRAM, SITE, SCRIPT, CONTENT MANAGEMENT SYSTEM, STRUCTURAL DIAGRAM.

The subject of the research is defining principles for building interactive web-based training courses based on CMS Drupal and PHP.

The object of study is interactive training course for students of computer science based on web technologies.

Purpose: to develop an interactive training course for teaching computer science to 7th class pupils based on web technologies.

The research method is a survey, analytical.

To accomplish this goal, the following tasks were solved:

- based on the analysis of the process of teaching computer science at school, the model of this process and the structure of the interactive educational system are proposed;

- methods and algorithms with probabilistic estimation of quantitative and qualitative parameters of educational system are developed;

- the choice of using modern tools to create interactive training courses on the technology of their use is substantiated;

- the effectiveness of the developed methodology was tested by creating and operating a training course based on integrated software.

MySQL database management system, Drupal content management system and PHP scripting language were used to perform the work.

ЗМІСТ

Завдання на кваліфікаційну роботу	2
Реферат	4
Summary	5
Вступ.....	7
1 Огляд області розробки та постановка задачі.....	9
1.1 Застосування інтерактивних навчальних курсів в освіті.....	9
1.2 Огляд існуючих навчальних курсів з інформатики для школярів .	12
1.3 Постановка задач дипломної роботи	19
1.4 Висновки до першого розділу.....	21
2 Вибір інструментів реалізації та розробка структури навчального курсу	22
2.1 Розробка структури інтерактивного курсу.....	22
2.2 Вибір методів та засобів реалізації навчального курсу.....	24
2.3 Розробка та опис структури бази даних	27
2.4 Висновки	37
3 Програмна реалізація навчального курсу	39
3.1 Створення інтерфейсу користувача.....	39
3.2 Заповнення навчальним контентом.....	60
3.3 Робота з додатковими та основними таблицями бази даних та створення додаткового функціоналу	63
3.4 Реалізація онлайн опитування рівня знань учнів.....	71
3.5 Реалізація автоматичної перевірки написаного учнями програмного коду	76
3.6 Висновки	79
Висновки.....	80
Перелік посилань	81
Додаток А Код файлу NewAnkete.php	83
Додаток Б Код файлу Results.php	87

ВСТУП

В умовах оновлення системи освіти особливої актуальності набуває проблема комп'ютеризації. Комп'ютер як засіб навчання займає провідне місце в освітньому процесі, по-перше, через прекрасних інформаційних, наочних та інших можливостей обчислювальної техніки і, по-друге, через специфічний інтегруючо-пізнавальної дії на учнів. Сучасне програмне забезпечення дозволяє побудувати процес навчання у вигляді розгалуженого інтерактивного процесу, різко підвищуючи увагу, ступінь засвоюваності, індивідуалізуючи навчання, розвиваючи самостійність в оволодінні навчальним матеріалом.

Основою сучасної політики уряду України у всіх економічних сферах є діджиталізація всіх процесів і в тому числі в галузі освіти. Тому важливим є розробка інтерактивних навчальних курсів з усіх навчальних предметів, як в середній школі так і в вищих навчальних закладах з використанням самих передових інформаційних технологій.

Мета роботи: розробити інтерактивний навчальний курс навчання інформатиці школярів 7 – го класу на базі web технологій.

Основні завдання роботи:

- на основі аналізу процесу навчання інформатики в школі запропонувати моделі цього процесу і структуру інтерактивної навчальної системи;
- розробити методи і алгоритми по ймовірнісної оцінкою кількісних і якісних параметрів навчальної системи;
- обґрунтувати вибір застосування сучасних засобів для створення інтерактивних навчальних курсів і освоїти технологію їх використання;
- перевірити ефективність розробленої методики шляхом створення і експлуатації навчального курсу на базі інтегрованих програмних засобів.

Для вирішення поставлених завдань були застосовані наступні загальноприйняті методи дослідження: теоретичні (аналіз педагогічної, психологічної та математичної літератури з проблеми дослідження, порівняння існуючих програмних засобів для реалізації процесу навчання); емпіричні (моделювання процесу навчання, моделювання функціонування навчального курсу, вимір кількісних і якісних показників роботи з інтерактивним навчальним курсом, отриманих експериментальним шляхом).

Методологічною основою дослідження стали педагогічні теорії організації процесу навчання в школі, теорії побудови інтерактивних навчальних курсів.

Теоретичну основу роботи складають: основи теорії організації програмованого навчання, комп'ютерного процесу навчання, дистанційного навчання, методика системного аналізу і моделювання процесів із застосуванням сучасних програмних засобів.

1 ОГЛЯД ОБЛАСТІ РОЗРОБКИ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Застосування інтерактивних навчальних курсів в освіті

Актуальним напрямом в області розробки освітніх технологій є методи інтерактивного навчання. Такі методи, які також називають дистанційними, успішно впроваджуються в практику.

Останнім часом набув поширення термін «інтерактивне навчання». Він означає навчання, засноване на активній взаємодії з суб'єктом навчання (провідним, учителем, тренером, керівником). По суті, воно являє один з варіантів комунікативних технологій: їх класифікаційні параметри збігаються. Інтерактивне навчання – це навчання з добре організованою зворотним зв'язком суб'єктів і об'єктів навчання, з двостороннім обміном інформацією між ними.

Впровадження інтерактивних форм навчання – одне з найважливіших напрямків вдосконалення підготовки студентів в сучасному професійному навчальному закладі. Основні методичні інновації пов'язані сьогодні із застосуванням саме інтерактивних методів навчання.

Поняття «інтерактивний» походить від англійського «interact» (inter – «взаємний», act – «діяти»). Інтерактивне навчання – це спеціальна форма організації пізнавальної діяльності. Вона має на увазі цілком конкретні і прогнозовані цілі. Одна з таких цілей полягає у створенні комфортних умов навчання, при яких студент або слухач відчуває свою успішність, свою інтелектуальну спроможність, що робить продуктивним сам процес навчання.

Актуальність даної теми полягає в тому, що в даний час в рамках реалізації пріоритетного національного проекту «Освіта» активно здійснюється впровадження інформаційних технологій в освітній процес. Зосередження сучасних технічних засобів навчання сприяє модернізації

навчально–виховного процесу, активізує розумову діяльність учнів, сприяє розвитку творчості педагогів, дозволяють проводити дистанційне навчання, розвивають систему безперервної освіти, тим самим підвищуючи ефективність освітнього процесу. У вітчизняних і зарубіжних виданнях комп'ютеризація навчального процесу розглядається як один з актуальних факторів організації навчання того чи іншого предмету. Новітні інформаційні технології в навчанні дозволяють активніше використовувати науковий та освітній потенціал провідних університетів та інститутів, залучати найкращих викладачів до створення курсів дистанційного навчання, розширювати аудиторію учнів. Незважаючи на те, що вже накопичено багатий досвід в області комп'ютерного навчання, багато викладачів з обережністю ставляться до можливості застосування комп'ютерних засобів навчання. Також, необхідно відзначити, що процес комп'ютеризація навчання стикається з низкою проблем. Процес впровадження інформаційної технології в навчання досить складний і вимагає глибокого осмислення. З одного боку, вони грають важливу роль в забезпеченні ефективності освітнього процесу, з іншого – може з'явитися проблема темпу засвоєння учнями матеріалу за допомогою комп'ютера, тобто проблема можливої індивідуалізації навчання.

Одним із пріоритетних напрямків процесу інформатизації сучасного суспільства є інформатизація освіти, що представляє собою систему методів, процесів і програмно–технічних засобів, інтегрованих з метою збору, обробки, зберігання, поширення і використання інформації в інтересах її споживачів. Мета інформатизації полягає в глобальній інтенсифікації інтелектуальної діяльності за рахунок використання нових інформаційних технологій: комп'ютерних і телекомунікаційних.

Інформаційні технології надають можливість:

– раціонально організувати пізнавальну діяльність учнів в ході навчального процесу;

- зробити навчання більш ефективним, залучаючи всі види чуттєвого сприйняття учня в мультимедійний контекст і озброюючи інтелект новим концептуальним інструментарієм;

- побудувати відкриту систему освіти, що забезпечує кожному індивіду власну траєкторію навчання;

- залучити до процесу активного навчання категорії дітей, що відрізняються здібностями і стилем навчання;

- використовувати специфічні властивості комп'ютера, що дозволяють індивідуалізувати навчальний процес і звернутися до принципово нових пізнавальних засобів;

- інтенсифікувати всі рівні навчально–виховного процесу [3].

Основні проблеми, що виникають при створенні інтерактивних навчальних курсів і організації комп'ютерного та дистанційного процесу навчання, такі:

- проведення системного аналізу процесу інтерактивного навчання, включаючи його моделювання і пропозиції щодо поліпшення;

- переробка навчального курсу для його комп'ютеризації;

- побудова навчального процесу із застосуванням комп'ютера;

- розробка засобів контролю знань, оцінювання рівня закріплення навичок і умінь;

- вибір інформаційних технологій для реалізації поставлених педагогічних і дидактичних завдань.

В Україні і в інших країнах світового співтовариства обговоренню підходів до вирішення цих проблем приділяється велика увага. Проводяться семінари, конференції та інші заходи з обміну досвідом у використанні комп'ютерної технології навчання в реальному навчальному процесі, де оцінюються позитивні і негативні сторони інформатизації освіти. Крім цього, є безліч журналів, присвячених також цієї тематики, наприклад, «Дистанційна освіта», «Інформаційні технології», «Інформатика та освіта», «Відкрита освіта». При цьому журнальні статті присвячені не тільки проблем

освіти середньої школи, а й дошкільних установ, середньо–спеціальної та вищої шкіл.

1.2 Огляд існуючих навчальних курсів з інформатики для школярів

Серйозна, орієнтована на результат підготовка в тому числі і з інформатики, по суті починається в 7–11–х класах. Однак, до цього моменту у дитини вже сформований арсенал основних «засобів навчання» і закладені основи його понятійного апарату. Дуже багато концепції на цьому етапі даються йому вкрай важко. Тому вкрай важливим є саме освоєння матеріалу в періоді 7го класу особливо з урахуванням того, що до цього все взаємодії з комп'ютером у учня відбувалися більше з точки зору користувача, а не творця чогось нового як це відбувається при навчанні інформатики починаючи з 7–го класу і вимагає самостійного написання програмного коду.

В процесі виконання роботи був проведений аналіз онлайн систем навчання інформатики або програмування.

Центр дистанційної освіти Ейдос

Найстаріший російський ресурс, створений в 1998 році. Платформою ресурсу є наукова школа А.В. Хутірського – засновника і директора Центру, доктора педагогічних наук.

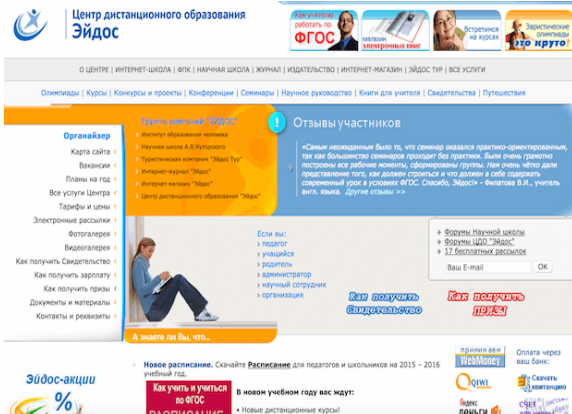


Рисунок 1.1 – Головна сторінка сайту Ейдос

Центр дистанційної освіти проводить відразу кілька курсів для школярів, починаючи з 5 – го класу.

Тут можна замовити розробку індивідуального курсу з цікавить вас, взяти участь в евристичних олімпіадах для школярів, подивитися уроки з інформатики та програмування в реальному часі.

Недоліки: російський, платний, відсутній швидкий контакт з вчителем, відсутня автоматична перевірка результатів тестів з бальною оцінкою, відсутня перевірка програмного коду та Взагалі форм для його написання.

«Фоксфорд»

Онлайн–школа «Фоксфорд» для учнів 5–11 х класів . За допомогою цього сервісу у дитини з будь–якого регіону Росії є можливість вчитися у кращих викладачів країни і надходити в кращі вузи.



Рисунок 1.2 – Головна сторінка сайту «Фоксфорд»

На ресурсі представлені наступні курси з програмування:

- мова Python, 8–10 х класів;
- java і Android, 9–11 х класів;
- мова Python для початківців, 6–8 х класів;
- введення в C / C ++, 8–11 х класів.

Заняття проходять на сайті в реальному часі. Можливо задавати питання через текстовий і голосовий чати. Доступні відеозапису занять для учасників.

Після кожного заняття відкривається доступ до домашнього завдання і конспекту по пройдених тем. При виконанні домашнього завдання можна використовувати інтерактивні підказки, звертатися до конспектів або записи минулого заняття.

Для навчання потрібен лише комп'ютер, ноутбук або iPad і підключення до Інтернету на швидкості від 0,7 Мбіт / с. Заняття платні.

Недоліки: російський, платний, відсутня автоматична перевірка результатів тестів з бальною оцінкою, відсутня перевірка програмного коду та взагалі форм для його написання.

Освітній Портал TeachPro

Перший «хмарний» освітній сервіс в Рунеті з мультимедійними інтерактивними електронними освітніми послугами, доступними з будь-якого браузера з будь-якою операційною системи на комп'ютері, планшеті або смартфоні.

TeachPro.ru с нами 23986 учеников, скачано 11197 курсов, подключены 302 школы

ПРИГЛАСИТЕ ДРУГА получите подарок

Иностранные языки Операционные системы Облачные сервисы Офисные программы Программирование Графические редакторы Школа ВУЗ Прочие

Начните обучение сегодня!

Более 250 интерактивных видеокурсов длительностью более 3500 часов!

[Начать обучение](#)

▶ → ! → ✓

Смотрите видеолекции
Вам доступно более 250 курсов!
Английский, Немецкий, Китайский, Java, Excel 2010, Photoshop CS6, Физика, ИЗО, ПДД, Шахматы, Маркетинг и др.

Обучайтесь в режиме контрольных вопросов
Это поможет закреплять полученные знания во время просмотра видеолекций

Выполняйте тестовые задачи
Проверяйте полученные знания, выполняя всевозможные интерактивные задания

Рисунок 1.3 – Головна сторінка сайту TeachPro

Сервіс із зрозумілим інтерфейсом. Легкий в управлінні. Курси програмування представлені в широкому діапазоні.

Основною особливістю сервісу є 100% мультимедійність – озвучені відео–та слайд–фільми, анімація, графіка. Діти можуть навчатися на онлайн–курсах з програмування з 7 років.

Недоліки: російський, платний, відсутній швидкий контакт з вчителем, відсутня автоматична перевірка результатів тестів з бальною оцінкою, відсутня перевірка програмного коду та взагалі форм для його написання.

Prometheus

Prometheus — це український проект масових відкритих онлайн–курсів, який був створений у 2014 році Іваном Примаченком та Олексієм Молчановським. Серед його партнерів найкращі виші країни: Києво–Могилянська академія, КНУ ім. Тараса Шевченка, Київський політехнічний інститут, Український католицький університет та Львівська ІТ–школа (LITS).

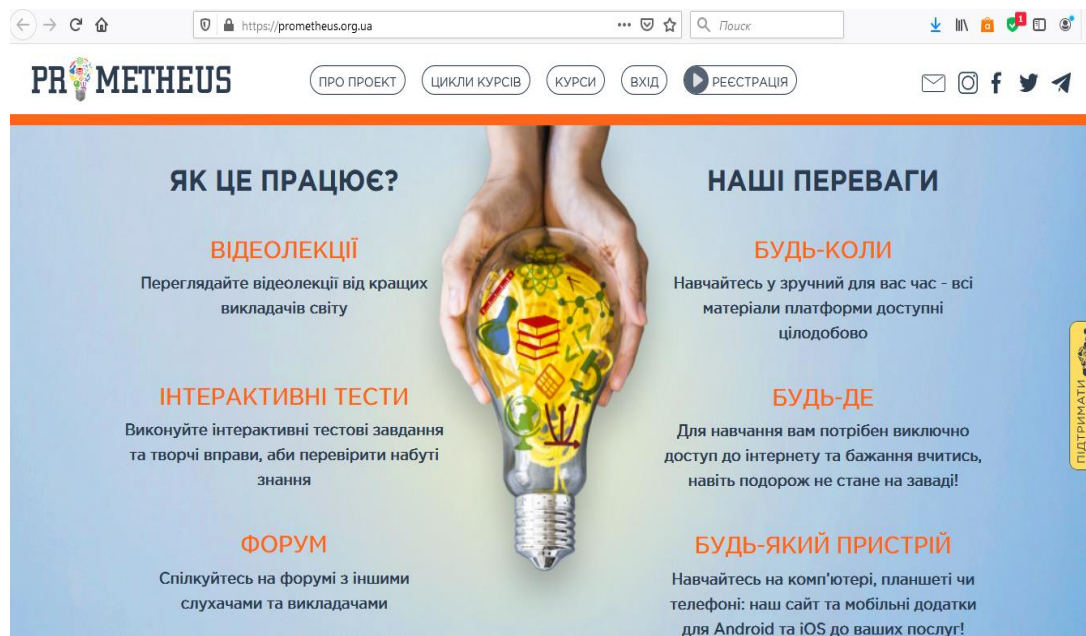


Рисунок 1.4 – Головна сторінка сайту Prometheus

Курси на платформі створено українською мовою, тож навчання доступне всім.

Курси на Prometheus безкоштовні й доступні в мережі у будь-який час. Є й мобільний додаток для Android та iOS, що дає змогу вчитися в дорозі або в будь-яку вільну хвилину за допомогою мобільних гаджетів.

Курси Prometheus побудовані за системою пов'язаних між собою циклів, наприклад, за темою аналізу даних, підприємництва чи громадянської освіти. Є й цикл курсів з підготовки до ЗНО. Тож користувач, який зацікавився, наприклад, підприємництвом чи програмуванням, може пройти не один окремий курс, а цілу низку.

Недоліки: рівень зависокий для учнів 7го класу, відсутній швидкий контакт з вчителем, відсутня перевірка програмного коду та взагалі форм для його написання.

ДистОсвіта

У дистанційних курсах сайту:

- навчальні матеріали: презентації, статті, уроки, посилання та відео-ролики відповідно до теми заняття; саме такими мають бути сучасні електронні підручники

- інтерактивні завдання, вправи та тренажери – ними можна скористатись як для класної, так і самостійної чи домашньої роботи

- практичні роботи із можливістю перевірки та коментування вчителем; переважна більшість практичних робіт містить різнорівневі та творчі завдання

- онлайн тестування з миттєвою перевіркою; тести містять запитання різних типів та мультимедійні об'єкти, вимагаючи не механічного відтворення, а креативного застосування знань

Всі курси сайту відповідають оновленим навчальним програмам, і дозволяють поєднувати окремі складові різних курсів для формування індивідуальних траєкторій навчання. Відповідно до змін у програмах, у курсах 5, 6, 7, 8, 10 класу визначені очікувані результати навчання: знаннєві, діяльнісні та ціннісні. До кожної теми додано анкету з обліку набуття компетентностей.

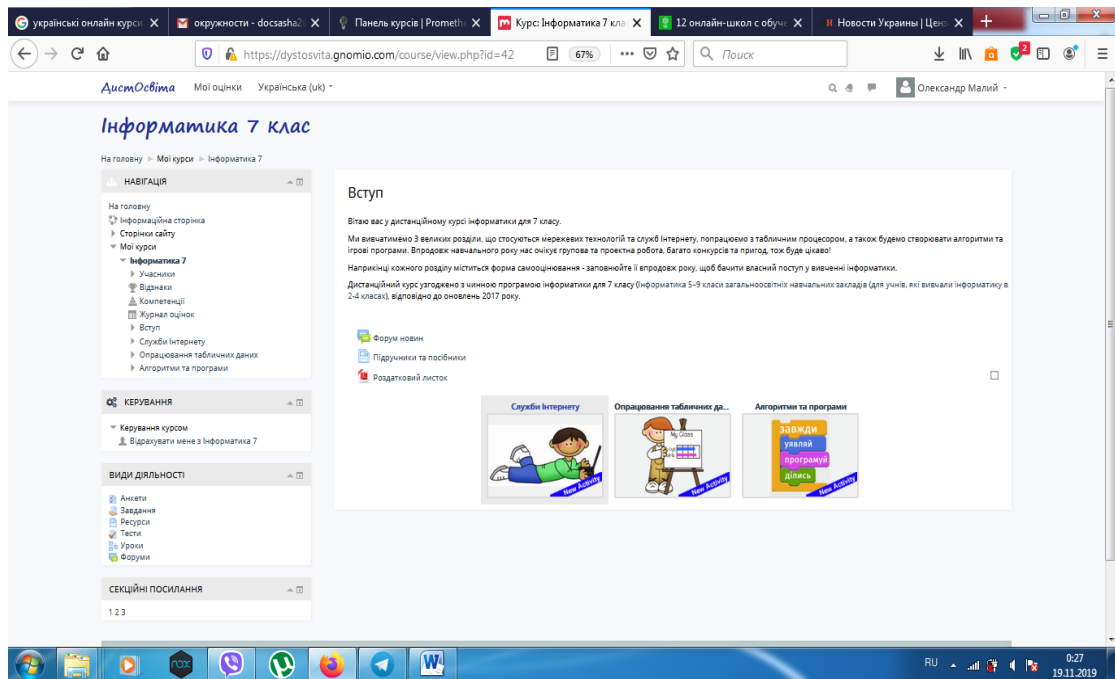


Рисунок 1.5 – Сторінка курсу інформатика 7 клас

Недоліки: відсутній швидкий контакт з вчителем, відсутня перевірка програмного коду та взагалі форм для його написання.

EdEra

Ще один український проект, який пропонує онлайн-освіту. Особливість EdEra у тому, що платформа пропонує багато курсів, які орієнтовані на вчителів: про викладання у початковій школі, недискримінаційні підходи в освіті, роботу з дітьми з особливими освітніми потребами тощо. Є й курси з української мови, літератури, історії, фізики та багато іншого. Журналістів може зацікавити курс з економічної журналістики, а правників — про захист прав людини. Для школярів на EdEra є окрема ЗНО-платформа. А в блозі команда платформи викладає поради про те, як ефективно конспектувати, вивчати та запам'ятовувати матеріали.

Серед партнерів EdEra Міністерство освіти і науки України, Освіторія, VoxUkraine, Intel, Studway та багато інших. Поки що на EdEra немає такої вражаючої кількості матеріалів, як на Coursera чи Prometheus.

Недоліки: відсутній швидкий контакт з вчителем, відсутня перевірка програмного коду та взагалі форм для його написання.

Зведений аналіз по основним показникам занесено в таблицю 1.1.

Таблиця 1.1 – Зведений аналіз онлайн систем навчання інформатиці

Назва ресурса	Оплата	Наявність курсу інформатики або програмування для школярів	Наявність зворотнього зв'язку з педагогом	Наявність онлайн тестів знань	Наявність автоматичної перевірки знань	Наявність автоматичної перевірки написаного
Ейдос	платний	Інформатика з 5го класу	+/-	+/-	+/-	-
«Фоксфорд»	платний	Python для початківців, 6–8 х класів	+	+/-	+/-	-
TeachPro	платний	Програмування з 7 років	-	+/-	+/-	-
prometheus	безкоштовний	Надвисокий для 7го класу	-	+	+/-	-
ДистОсвіта	безкоштовний	Інформатика з 5го класу	-	+/-	+/-	-
EdEra	безкоштовний	Інформатика з 5го класу	-	+/-	+/-	-

Загалом провівши аналіз можна сказати, що в інтернеті представлено дуже мало українських навчальних курсів, та всі вони як правило або мають погані можливості зворотнього зв'язку з вчителем або в них відсутня автоматична перевірка знань. Крім того жодні онлайн курси не дозволяють проводити автоматичну перевірку написаного учнем коду.

1.3 Постановка задач дипломної роботи

Аналіз існуючих аналогів показав, що існуючі курси представлені в Україні як правило є електронною копією підручників іноді з введенням тестової перевірки результатів проміжних знань отриманих на кожному окремому етапі (при вивченні окремої теми). Не в одному курсі не запропоноване повністю автоматична перевірка якості знань в області написання програмного коду, а також слабо реалізований або взагалі не реалізований процес комунікації з викладачем (учителем) і учасниками групи учнів.

Крім того – навчальний матеріал, особливо на першій половині курсу слабо пов'язаний з практичними роботами учня безпосередньо на комп'ютері, що в свою чергу сильно послаблює інтерес учня до отримання нових знань.

Учні 7-го класу – це ще особистості, які формуються з дітей у дорослих і процес піднесення їм знань повинен проходити з елементами практичних занять які підтверджують запропоновану теорію. Крім того даний процес повинен бути побудований з елементами змагальності і в напівігровий формі.

У зв'язку з цим в дипломному проєкті крім використання стандартної шкільної програми з інформатики пропонується використання програмно-апаратної платформи Ардуіно на основі якої учень зможе відразу бачити нехай найпростіші, але все ж результати своїх досягнень.

В даному курсі, що розробляється в дипломному проекті, процес вивчення інформатики для школярів проводиться в два етапи, на першому даються основні поняття цифрової обробки інформації за допомогою комп'ютера, на другому з використанням платформи Ардуіно. Це проводиться з метою посилення уваги і бажання навчатися інформатиці, оскільки на платформі Ардуіно можна не тільки навчитися програмувати, а й збирати цікаві робототехнічні конструкції які школяр може створювати самостійно, що як правило, привертає додатковими інтерес до знань. Для того щоб робота навчального курсу не зависла від поточної доступності викладача (вчителя) в мережі курс пропонується виконати максимально автоматизованим.

Крім того курс заснований на Arduino, тому що вона проста у вивченні і в використанні. Також до цієї платі існує велика кількість вже готових модулів, і завжди є можливість створити свої власні.

Особливостями розробки в дипломному проекті інтерактивного навчального курсу є:

- максимально наближеній до рівня знань дітей 7го класу виклад матеріалу (складні речі подаються в простій формі з прикладами і аналогіями з життя);
- присутність розділу пошуку відповідей на найбільш затребувані питання по кожному уроку;
- реалізований механізм зворотного зв'язку з викладачем (учителем) як за допомогою електронної пошти, так і за допомогою поширених меседжер;
- існує автоматична перевірка тестових завдань в онлайн режимі, що дозволить учню відразу перевірити свої знання;
- використання програмно–апаратної системи Ардуіно дозволить учню швидко отримувати власні результати від використання свіжих знань в області програмування і робототехніки;

– існує візуалізація роботи розробленої відповідно до завдання програми за рахунок можливості віддаленого управління платформою Ардуіно.

Можливості платформи Arduino

Цей процесор дозволяє виконувати різні функції:

– отримання інформації з різних датчиків температури, світла, звуку, відстані, обробка інформації з них;

– управління моторами, їх програмування, завдання параметрів роботи;

– можливість дистанційного керування пристроями за допомогою Bluetooth або мережі Інтернет.

У процесі навчання, учням стають доступні основи мови програмування C ++, а на практиці ці знання будуть успішно закріплюватися. Онлайн курс з інформатики стає дуже зручним способом отримання інформації – дітям не потрібно нікуди їздити.

1.4 Висновки до першого розділу

У розділі проведено огляд необхідності застосування інтерактивних освітніх курсів особливо для школярів при вивченні навчальних предметів. Також було розглянуто існуючі аналоги наявних інтернет ресурсів, що дозволяють проводити навчання інформатиці та програмуванню. Аналіз показав, що жоден з наявних курсів не дозволяє проводити автоматичну перевірку програмного кода та не задіює меседжери для швидкої взаємодії учня з вчителем. Наприкінці пункту було поставлено задачі кваліфікаційної роботи.

2 ВИБІР ІНСТРУМЕНТІВ РЕАЛІЗАЦІЇ ТА РОЗРОБКА СТРУКТУРИ НАВЧАЛЬНОГО КУРСУ

2.1 Розробка структури інтерактивного курсу

Згідно запропонованої концепції навчання була запропонована наступна структура роботи інтерактивного навчального курсу (рис. 2.1).

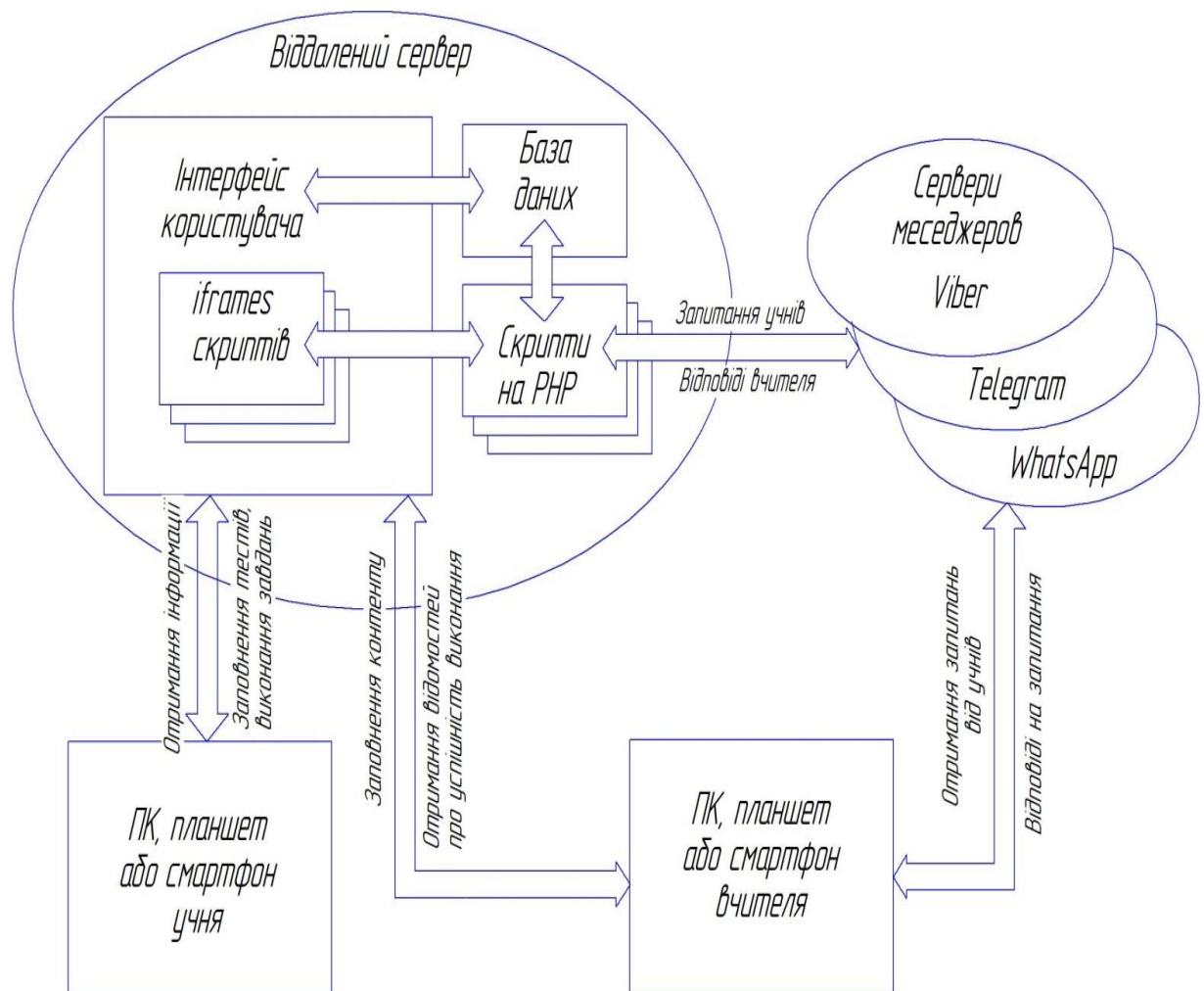


Рисунок 2.1 – Структура навчального курсу

Навчальний курс має графічно– текстове оздоблення, що має вигляд інтерфейсу користувача, що відображається при підключенні на віддалений сервер шляхом введення адреси інтернет сторінки. Реалізувати такий інтерфейс можна достатньою кількістю інструментів. Вибір окремо для

наших потреб ми розглянемо у наступному пункті. Інтерфейс користувача буди собою являти різні графічно–текстові блоки з інтуїтивно зрозумілим меню та можливість авторизуватися для розрізнення рівня користувачів. Пропонується зробити три рівні користувачів:

- учень;
- вчитель;
- адміністратор сайту.

Інтерфейс користувача зберігає дані щодо зовнішнього вигляду та змісту сторінок у базі даних (це найросповсюдженіший алгоритм зберігання даних на віддалених серверах). Оскільки у нас використовується не лише просто відображення навчального матеріалу, а й присутні нестандартні функції по обробці даних (перш за все тестування з автоматичною перевіркою, та задачі по програмуванню, які передбачають автоматичний аналіз введених даних) на сервері також будуть реалізовуватись окремі скрипти, що будуть обробляти нестандартну інформацію. Ці скрипти будуть відображатися на інтерфейсі користувача у вигляді фрагментів iframes та також зберігати свою інформацію у свої окремих таблицях бази даних.

При заповненні змісту курсу вчитель входить на адресу сайту курсу, авторизується та у відповідному розділі заповнює мета ріали курсу, створює тести, правила їх оцінювання, створює завдання, вказує налаштування для зворотнього зв'язку та може переглядати результати.

У свою чергу учень з'єднується з віддаленим сервером шляхом введення адреси інтернет сайту, авторизується на ньому та починає вивчати запропоновані вчителем теми та по проходжені тем виконувати завдання. При цьому більша частина завдані перевіряються на сервері автоматично та учню повідомляється успішність його з тієї чи іншої теми. Для розвитку змагальної заохочуваності можливо використання загально доступної статистики серед учнів як у відкритій формі (з вказанням прізвищ), так і закритій (вказання загальної кількості учнів, що вже виконали завдання на визначений бал). Також учень може задавати запитання вчителю.

Оскільки курс, що розробляється націлений на швидкий та по можливості автоматизований процес перевірки знань у якості зв'язку з викладачем (вчителем) було обрано меседжери – оскільки просто листи електронної пошти вчитель може побачити/перевірити не одразу, а повідомлення в меседжері прийдуть відразу на телефон чи інший пристрій де встановлено відповідний додаток та вчитель, у разі якщо запитання просте може відразу відповісти. Відправлення повідомлень пропонується реалізовувати через сервер, щоб безпосередньо не давати прямий номер вчителя учню, що дозволить в разі наявності свобідного часу відповісти відразу на поставлене питання, а при високій зайнятості написати відповідь через якийсь час. Крім того використання такого механізму при подальшому розвитку курсу надасть можливості автоматичної відповіді на питання, що часто повторюються без відправлення питання вчителю (так звані боти для меседжерів). У системі було обрано три самих розповсюджених меседжера (Viber, Telegram WhatsApp), при цьому вчителю пропонується обрати як лише один з них так і декілька відразу (при цьому повідомлення будуть дублюватися сервером на усі обрані меседжери) та вчитель зможе відповісти на будь якому. Крім того можливо використання меседжерів для отримання статистики успішності учнів (скільки тем пройшли та які бали отримали при виконанні завдань).

2.2 Вибір методів та засобів реалізації навчального курсу

Оскільки навчальний курс з боку учня буде представляти із себе інтернет сайт з графічно–текстовим врученням матеріалу і можливістю вставки відео і звуку для реалізації інтерфейсу користувача було прийнято рішення використовувати стандартизовані інструменти створення графічного образу.

Щоб спростити процес створення графічного інтерфейсу існують два основні підходи:

- використання FrameWork систем;
- використання систем управління вмістом (CMS – content management systems).

Фреймворк від англійського слова "каркас". Фреймворк – це надбудова над мовою, набір бібліотек. Добре налагоджених і прекрасно взаємодіють між собою бібліотек

CMS – це, по суті, вже готовий сайт, який потрібно налаштувати під свої потреби.

CMS значно прискорює розробку простого шаблонного сайту. У сайту відразу готова админка і її не треба писати окремо, на відміну від розробки на фреймворку. Однак це швидкість створення сайту досягається за рахунок шаблонності, обмеженості або зайвої універсальності CMS.

У фреймворк часто тільки самий необхідні функції і хороший, налагоджений, отрефакторений код.

Оскільки в дипломному проекті передбачається досить велика кількість вмісту (навчальних матеріалів), а механізми авторизації і створення розділів меню і блоків стандартні було прийнято рішення зупинитися на другому підході –використання CMS.

Популярні CMS WordPress, Joomla, Drupal, Magento, Bitrix, Prestashop, MODX, OpenCart. Всі ці CMS написані на PHP. CMS в основному пишуться на php через те, що їх можна встановити де завгодно, хоч на найдешевшому хостингу.

Одні з кращих представників CMS:

- Drupal – характеризується як гнучкий інструмент, оснащений відмінною документацією і супроводжується спільнотою фахівців;
- Joomla – затребувана, але мало ефективна в порівнянні з іншими CMS;

– WordPress – відмінна система для створення блогів, сайтів візиток або новинних сайтів.

Системою управління вмістом була обрана CMS Drupal.

Drupal – безкоштовний движок, який можна завантажити з сайту розробника або встановити через панель адміністратора на хостингу. Другий спосіб краще, оскільки при його використанні немає необхідності створювати базу даних і виконувати попередню настройку системи – досить прикріпити до аккаунту на хостингу зареєстроване доменне ім'я і вибрати його для розгортання системи управління контентом.

При установці Drupal вже має стандартний комплект модулів, за допомогою яких можна керувати сайтом. Серед них:

- Tracker для відстеження нових товарів.
- Block для виведення контенту на сторінках.
- Color для змін колірної схеми сайту, якщо таку можливість надає тема оформлення.
- Image для обробки зображення.
- Node для додавання нових матеріалів.
- Search для організації пошуку по сайту.
- System для управління системними параметрами.
- User для управління зареєстрованими користувачами.
- Locale для вибору мовних пакетів.
- Path для перейменування посилань.
- Update Manager для перевірки оновлень.

Бази даних для зберігання інформації будуються на підставі СУБД.

Оскільки в якості базової системи створення призначеного для користувача інтерфейсу була обрана CMS Drupal яка створює базу даних MySQL, а також всі поширені хостинг провайдери в своїх інструментах мають дану СУБД – було прийнято рішення використовувати саме її.

MySQL – вільна реляційна система управління базами даних.

Гнучкість СУБД MySQL забезпечується підтримкою великої кількості типів таблиць: користувачі можуть вибрати як таблиці типу MyISAM, що підтримують повнотекстовий пошук, так і таблиці InnoDB, що підтримують транзакції на рівні окремих записів. Більш того, СУБД MySQL поставляється із спеціальним типом таблиць EXAMPLE, що демонструє принципи створення нових типів таблиць. Завдяки відкритій архітектурі і GPL-ліцензуванню, в СУБД MySQL постійно з'являються нові типи таблиць.

В якості мови програмування для написання скриптів специфічних функцій була вибрана мова PHP, в першу чергу це пов'язано з тим що більшість сайтів реалізуються саме на цій мові програмування, CMS Drupal написана на PHP і мову PHP дозволяє створювати запити до бази даних MySQL як простими конструкціями з використанням прямих SQL запитів так і з використанням бібліотек і структурованих масивів.

2.3 Розробка та опис структури бази даних

Основну частину бази даних при установці CMS Drupal створює автоматично при установці на сервер. В основну частину включаються таблиці пов'язані з авторизацією користувачів, зовнішнім оформленням блоків, вмістом сторінок і їх адресами.

При установці CMS Drupal створює таблиці бази даних наведені нижче (в загальному 47 таблиць).

- а) таблиця batch – таблиця пакетів даних користувача;
- б) таблиці, що відповідають за розташування та наповнення блоків сайту: block_content, block_content_field_data, block_content_field_revision, block_content_revision, block_content_revision__body, block_content__body;
- в) таблиці для обробки кешу сайту: cachetags, cache_bootstrap, cache_config, cache_container, cache_data, cache_default, cache_discovery,

cache_dynamic_page_cache, cache_entity, cache_menu, cache_page, cache_render;

г) таблиці для зберігання та обробки коментарів: comment, comment_entity_statistics, comment_field_data, comment__comment_body;

д) таблиця налаштувань вибору теми відображення блоків config;

е) таблиці для роботи з файлами: file_managed, file_usage;

ж) таблиця history – історія відвідування користувачем сторінок сайту;

з) таблиці по роботі з ключами користувачів при виконанні операції редагування сайту key_value, key_value_expire;

и) таблиці по роботі з підключеними модулями: CMS locales_location, locales_source, locales_target, locale_file;

к) таблиці по роботі з меню: menu_link_content, menu_link_content_data, menu_tree;

л) таблиці по роботі з сторінками (вузлами) сайту: node, node_access, node_field_data, node_field_revision, node_revision, node_revision__body, node_revision__comment, node_revision__field_image, node_revision__field_tags, node__body, node__comment, node__field_image, node__field_tags;

м) таблиця queue – робота з чергами запитів.

Оскільки у нас в курсі, згідно розробленої структури, реалізується додатковий функціонал, що покликаний проводити автоматичне тестування знань учнів, який реалізовується на PHP скриптах, додатково створюються таблиці бази даних завдання яких містити інформацію про тести, а також результати їх заповнення учнями з можливістю перевірки правильності заповнення. Дані таблиці наведені нижче (табл. 2.1 – 2.9).

Таблиця 2.1 – Структура таблиці БД Autors (облікові дані вчителя)

№	Назва стовбця	Тип даних	Опис даних, що зберігаються	На які таблиці посилається

Продовження таблиці 2.1

1	Id	INT	Первинний ключ	
2	_login	varchar(50)	Логін користувача	
3	password_md5hash	varbinary(max)	Хеш паролю	
4	facebook_acc	varchar(max)	Посилання на акаунт в соціальній мережі Facebook	
5	access_level	INT	Рівень доступу	
6	first_name	varchar(50)	Ім'я автора (вчителя)	
7	last_name	varchar(50)	Прізвище автора (вчителя)	

Стовбець id зберігає унікальний цілочисельний ідентифікатор запису.

Стовбець _login зберігає ім'я користувача (автора/вчителя) довжиною до 50 символів (бажано, для запобігання не правильного кодуванню, задавати значення латинськими буквами, можливо з використанням чисел).

Стовбець password_md5hash зберігає хеш паролю користувача (автора/вчителя). Використовується саме хеш, а не пароль у чистому вигляді, оскільки це дає змогу захистити пароль від прямого перегляду.

Паролі користувачів саме хешируються, а не шифруються з наступних причин.

а) трудомісткість. Шифрування може тривати довше, а яке перетворення ми б не вибрали, його доведеться проробляти при кожній перевірці пароля. Однією з вимог до хеш-функцій ж є швидкість виконання;

б) довжина вихідних значень. Результат шифрування має змінну довжину, результат хешування – завжди однаковою, а зберігати однорідні за

розміром дані в базі даних дуже вже зручно. Не кажучи вже про те, що довжина пароля в зашифрованому вигляді буде давати деяку інформацію про довжину вихідного пароля. Однакова довжина, правда, призводить до можливості виникнення колізій, але про це нижче;

в) управління ключами. Для шифрування потрібно ключ, який теж десь доведеться зберігати і сподіватися, що його ніхто не знайде. У будь-якому випадку, генерація і управління ключами це окрема історія (вони не повинні бути слабкими, їх потрібно регулярно міняти і так далі).

Hash (хеш функція) – функція однозначного відображення рядка (будь-якої довжини) на кінцеву множину (рядок заданої довжини).

Саме число (рядок) хеш – результат обчислення хеш-функції над даними.

Існують криптографічні та некриптографічні (класифікуються окремо, до них відносяться, наприклад, контрольні суми) хеш-функції.

Для криптографічних хеш є три додаткових умови, які відрізняють їх від всіх інших:

а) незворотність: для заданого значення хеш-функції m повинно бути обчислювально нездійсненно знайти блок даних X , для якого $H(X) = m$;

б) стійкість до колізій першого роду: для заданого повідомлення M має бути обчислювально нездійсненно підібрати інше повідомлення N , для якого $H(N) = H(M)$;

в) стійкість до колізій другого роду: має бути обчислювально нездійсненно підібрати пару повідомлень $\sim (M, M')$, що мають однаковий хеш.

Стовбець `facebook_acc` – посилання на акант у соціальній мережі facebook для можливості входу посиланням без введення паролю (можливо лише після реєстрації з пов'язуванням аканту з соціальною мережею).

Стовбець `access_level` – зберігає рівень доступу користувача до редагування анкет (учень, вчитель, адміністратор системи).

Стовбець `first_name` – власне ім'я автора/вчителя до 50 символів (не обов'язково співпадає з логіном).

Стовбець `last_name` – прізвище автора/вчителя до 50 символів.

Таблиця 2.2 – Структура таблиці БД Users (облікові данні учня)

№	Назва стовбця	Тип даних	Опис даних, що зберігаються	На які таблиці посилається
1	<code>id</code>	INT	Первинний ключ	
2	<code>_login</code>	<code>varchar(50)</code>	Логін користувача	
3	<code>password_md5hash</code>	<code>varbinary(max)</code>	Хеш паролю	
4	<code>facebook_acc</code>	<code>varchar(max)</code>	Посилання на акаунт в соціальній мережі Facebook	
5	<code>access_level</code>	INT	Рівень доступу	
6	<code>first_name</code>	<code>varchar(max)</code>	Ім'я учня	
7	<code>last_name</code>	<code>varchar(max)</code>	Прізвище учня	
8	<code>idAutor</code>	INT	id автора (вчителя)	<code>Autors.id</code>

Стовбці цієї таблиці майже повністю співпадають з попередньою з різницею в тому, що тут зберігаються дані щодо учнів та вони відносяться до свого окремого вчителя, посилання на якого зберігається у стовбці idAutor.

Таблиця 2.3 – Структура таблиці БД Questionnaires (дані об анкетах)

№	Назва стовбця	Тип даних	Опис даних, що зберігаються	На які таблиці посилається
1	Id	INT	Первинний ключ	
2	Name	varchar(250)	Назва анкети	
3	idAutor	INT	id автора	Autors.id
4	Create_Date	Date	дата та час створення	
5	QCount	INT	Максимальна кількість заповнених анкет	

Стовбець id зберігає унікальний цілочисельний ідентифікатор запису.

Стовбець Name – назва анкети (до 250 символів). Назва анкети може

Стовбець idAutor – числове посилання на автора анкети/вчителя, що створив цю анкету.

Стовбець Create_Date – зберігає дату та час створення анкети.

Стовбець QCount – Зберігає максимальну кількість анкет яку може заповнити один учень для того щоб кількість спроб не була нескінченною і можна було б адекватно оцінити швидкість навчання. Можна також проводити оцінювання не лише на основі останньої спроби, а й враховувати кількість спроб для отримання позитивного результату.

Таблиця 2.4 – Структура таблиці БД Res_Questionnaires (дані стосовно заповненої анкети)

№	Назва стовбця	Тип даних	Опис даних, що зберігаються	На які таблиці посилається
1	id	INT	Первинний ключ	
2	idQuestionnaires	INT	id Анкети	Questionnaires (id)
3	idUser	INT	id учня	Users (id)
4	Create_Date	datetime	Дата та час заповнення анкети	
5	GPS	varchar(max)	Координати GPS	

Стовбець id зберігає унікальний цілочисельний ідентифікатор запису. Кількість записів таблиці відповідає кількості повністю заповнених анкет.

Стовбець idQuestionnaires зберігає посилання на ідентифікатор анкети яку було заповнено.

Стовбець idUser зберігає посилання на ідентифікатор користувача (учня), який заповнив анкету.

Стовбець Create_Date зберігає дату та час закінчення заповнення анкети

Стовбець GPS зберігає координати де було заповнено анкету, для виявлення, можливості заповнення не на території окремо відведених даних.

Таблиця 2.5 – Структура таблиці БД Texts (тексти питань та відповідей)

№	Назва стовбця	Тип даних	Опис даних, що зберігаються	На які таблиці посилається
1	id	INT	Первинний ключ	

Продовження таблиці 2.5

2	text_var	text	Текст	
---	----------	------	-------	--

Стовбець text_var зберігає всі тексти анкети. Тексти винесені в окрему таблицю, а в питаннях та відповідях на тексти зроблено лише посилання оскільки тексти можуть повторюватись.

Таблиця 2.6 – Структура таблиці БД Questions (дані о питаннях)

№	Назва стовбця	Тип даних	Опис даних, що зберігаються	На які таблиці посилається
1	id	INT	Первинний ключ	
2	idQuestionnaire	INT	id анкети	Questionnaires (id)
3	Quest_type	INT	Тип питання 0 – питання з фіксованою відповіддю, 1 – питання зі своїм варіантом відповіді	
4	Group_type	INT	Тип групування 0 – звичайне питання, 1 – стаття, 2 – вік, 3 – територія	
5	idText	INT	id тексту	Texts (id)
6	Answ_count	INT	Кількість одночасних відповідей в питанні	

Стовбець `idQuestionnaires` зберігає посилання на ідентифікатор анкети до якої відноситься це питання.

Стовбець `Quest_type` зберігає числовий тип питання. Питання можуть розділятися на питання за деякою кількістю фіксованих відповідей (індекс 0) та питання з відкритою відповіддю (індекс 1).

Стовбець `Group_type` зберігає номер типу групи питання. Питання можуть бути характеристиками статі та віку учнів, або території та класу де проводиться опитування – у цьому випадку ці питання будуть впливати на загальну статистику але не будуть впливати на оцінку за тестування.

Стовбець `idText` зберігає індекс тексту питання. Сам текст зберігається в таблиці `Texts`.

Стовбець `Answ_count` зберігає кількість відповідей, що можуть бути обрані одночасно. Звісно ця кількість не має перевищувати загальну кількість відповідей у питанні і не може бути менша за одиницю.

Таблиця 2.7 – Структура таблиці БД `Res_Questions` (питання, що заповнені в заповнених анкетах)

№	Назва стовбця	Тип даних	Опис даних, що зберігаються	На які таблиці посилається
1	<code>id</code>	INT	Первинний ключ	
2	<code>idRes_Questionnaire</code>	INT	Посилання на анкету, що заповнена	<code>Res_Questionnaires(id)</code>
3	<code>idQuestion</code>	INT	Посилання на оригінальне питання	<code>Questions(id)</code>

Ця таблиця (табл. 2.7) зберігає результати заповнення питання з індексом `idQuestion` для заповненої анкети з індексом `idRes_Questionnaire`.

Таблиця 2.8 зберігає дані про відповіді.

Стовбець idQuestion зберігає індекс питання до якого відноситься ця відповідь.

Стовбець answer_type зберігає номер типу відповіді. Дорівнює 0 якщо відповідь з фіксованим текстом, що надається вчителем, та дорівнює 1 якщо текст відповіді вводиться учнем самостійно.

Стовбець idText зберігає індекс тексту відповіді. Сам текст зберігається в таблиці Texts.

Таблиця 2.8 – Структура таблиці БД Answers (дані про відповіді)

№	Назва стовбця	Тип даних	Опис даних, що зберігаються	На які таблиці посилається
1	Id	INT	Первинний ключ	
2	idQuestion	INT	Посилання на питання	Questions(id)
3	answer_type	INT	Тип відповіді. 0 – фіксована відповідь 1 – відкрите питання, відповідь вводиться самостійно	
4	idText	INT	Посилання на текст	Texts (id)

Таблиця ResTexts (табл. 2.9) відповідає за зберігання власних відповідей коли відповіді вводяться самостійно учнем.

Таблиця 2.9 – Структура таблиці БД ResTexts (зберігання власних варіантів відповідей)

№	Назва стовбця	Тип даних	Опис даних, що зберігаються	На які таблиці посилається
1	Id	INT	Первинний ключ	

Продовження таблиці 2.9

2	text_var	varchar(max)	Текст	
---	----------	--------------	-------	--

Таблиця Res_Answers (табл.2.10) зберігає посилання на заповнене питання idRes_Question заповненої анкети з вказанням вибраного варіанту фіксованої відповіді idAnswer, або посилання на текст власної відповіді idText.

Таблиця 2.10 – Структура таблиці БД Res_Answers (відповіді на питання в заповненій анкеті)

№	Назва стовбця	Тип даних	Опис даних, що зберігаються	На які таблиці посилається
1	id	INT	Первинний ключ	
2	idRes_Question	INT	Посилання на питання в заповненій анкеті	Res_Questions(id)
3	idAnswer	INT	Посилання на оригінальну відповідь у анкеті	Answers(id)
4	idText	INT	id тексту в таблиці ResTexts, якщо відповідь має власний текст, якщо ні – то null	ResTexts(id)

2.4 Висновки до другого розділу

У другому розділі роботи було проведено розробку структури навчального курсу та описано всі його елементи та взаємодії між ними. Також було проведено вибір методів та засобів реалізації і обрано у якості інструментів реалізації CMS Drupal та скрипкову мову PHP. Оскільки в

навчальному курсі розробляється одатковий функціонал, який не дозволяє реалізувати система керування вмістом, то у розділі було також описано додаткові таблиці бази даних, що дозволять проводити тестування знань учнів.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ НАВЧАЛЬНОГО КУРСУ

3.1 Створення інтерфейсу користувача

Реєстрація хостингу та домену

При створенні сайту першим кроком є вибір хостингу для розміщення файлів даних та бази даних та вибір доменного ім'я (адресу сторінки) сайту.

Хостинг (англ. Hosting) – послуга з надання ресурсів для розміщення інформації на сервері, що постійно перебуває в мережі (зазвичай Інтернет). Зазвичай хостинг входить в пакет з обслуговування сайту і має на увазі як мінімум послугу розміщення файлів сайту на сервері, на якому запущене ПЗ, необхідне для обробки запитів до цих файлів (веб-сервер). Як правило, в обслуговування вже входить надання місця для поштової кореспонденції, баз даних, DNS, файлового сховища на спеціально виділеному файл-сервері і т. П., А також підтримка функціонування відповідних сервісів.

Доменне ім'я – символічне ім'я, що служить для ідентифікації областей, які є одиницями адміністративної автономії в мережі Інтернет, в складі вищої по ієрархії такої області. Кожна з таких областей називається доменом. Доменні імена дають можливість адресації інтернет – вузлів і розташованим на них мережевих ресурсів бути представленими в зручній для людини формі.

В якості хостінг провайдера було обрано сервіс Ukraine.com.ua (рис. 3.1) оскільки це український надійний сервіс з багаторічною практикою перевірений на протязі великої кількості часу та який має помірні ціни в порівнянні з закордонними сервісами такого рівня.

Після вибору хостингу необхідно обрати тарифний план, що підійде для розміщення сайту навчального курсу.

Оскільки навантаження на сайт навчального курсу буде не дуже велике та кількість матеріалу навчального курсу не дуже велика – було обрано

самий простий тариф представлений на сервері хостингу. Параметри обраного тарифного плану наведені у таблиці 3.1.

Таблиця 3.1 – Параметри обраного тарифного плану хостингу

Параметр	Значення	Параметр	Значення
Загальні параметри		PHP	
Місце на SSD диску	3ГБ	Версії, що підтримуються	4.4–7.4
SSL сертифікат	Безкоштовний	Повноцінний режим PHP без safe mode	Присутній
Сайтів	1	Підтримка CURL	Присутня
Ліміт оперативної пам'яті	512МБ	Zend Optimizer	Присутній
Кількість субдоменів	Необмежено	ionCube Loader	Присутній
MySQL		Phalcon	2,3
Кількість баз даних	Необмежено	Memcache	Присутній
Створення тригерів, view, процедур і так далі	Присутній	Redis	Присутній
Віддалене підключення до MySQL	Присутнє	Підтримка HTTP/2	Присутня
Доступ к phpMyAdmin	Присутнє		

The screenshot shows a web hosting service interface. At the top, there are navigation links: Хостинг, Регистрация доменов, Бизнес-хостинг, VPS, Выделенные сервера, FAQ, Форум, and Карьера. Below this is a search bar for domains with a 'Проверить' button. A list of domain names and their prices is displayed, including wiki, best, shop, xyz, online, space, club, site, tech, design, website, store, com, ink, and fun. Below the domain list, there are sections for 'Хостинг Украина' with buttons for 'Трансфер домена', 'Как оформить заказ', and 'Наши преимущества'. A 'Новости' section shows dates from 28 ноября. The main part of the screenshot is a table titled 'Стоимость хостинга - сравнение тарифных планов' comparing two plans: 'Сайт' (Basic plan for small sites) and 'Лучший' (Best for multiple sites). The table compares features like SSD disk space, SSL certificate, number of sites, memory limit, and subdomains.

	Сайт Базовый план для небольших сайтов	Лучший Оптимально для нескольких сайтов
Место на SSD диске	3 Gb	10 Gb
Бесплатный SSL сертификат	✓	✓
Сайтов	1	5
Memory limit (RAM)	512 Мб	1024 Мб
Субдоменов	неограниченно	неограниченно

Рисунок 3.1 – Вибір тарифного плану на хостингу

Після вибору тарифного плану хостингу обираємо доменне ім'я сайту. Обираємо ім'я robokurs.site оскільки домен другого рівня robokurs характеризує наявність навчання елементам робототехніки в рамках інформатики, а домен першого рівня site характеризує, що це буде сайт з цього приводу та це ім'я має невелику щорічну оплату.

Наступним етапом є встановлення системи керування контентом (CMS Drupal) на сервері.

Для встановлення на сервері CMS Drupal в нашому випадку можна використати 2 способи:

- встановлення з офіційного сайту drupal.org;
- встановлення інструментами хостингу.

У першому випадку необхідно виконати наступні дії по черзі:

а) зайти на <https://www.drupal.org> и перейти к пункту Build > Download & Extend в верхньому меню;

The screenshot shows the Drupal.org website. At the top, there is a navigation menu with links: Build, Solutions, Services, Community, Resources, Give, and Try Drupal. Below the menu, the main heading is 'Download Drupal'. Underneath, there is a sub-heading: 'Drupal allows you to create a unique space in a world of cookie-cutter solutions.' Below this, there is a prominent blue button labeled 'Download Drupal zip'. Underneath the button, there are two links: 'download tar.gz' and 'read release notes'.

Рисунок 3.2 – Посилання для скачування на офіційному сайті

- б) завантажити файл в архіві та зберегти на комп'ютері;
 в) за допомогою FTP панелі хостингу або за допомогою будь якого FTP менеджера завантажити отриманий архів в папку сайта на сервері хостингу.

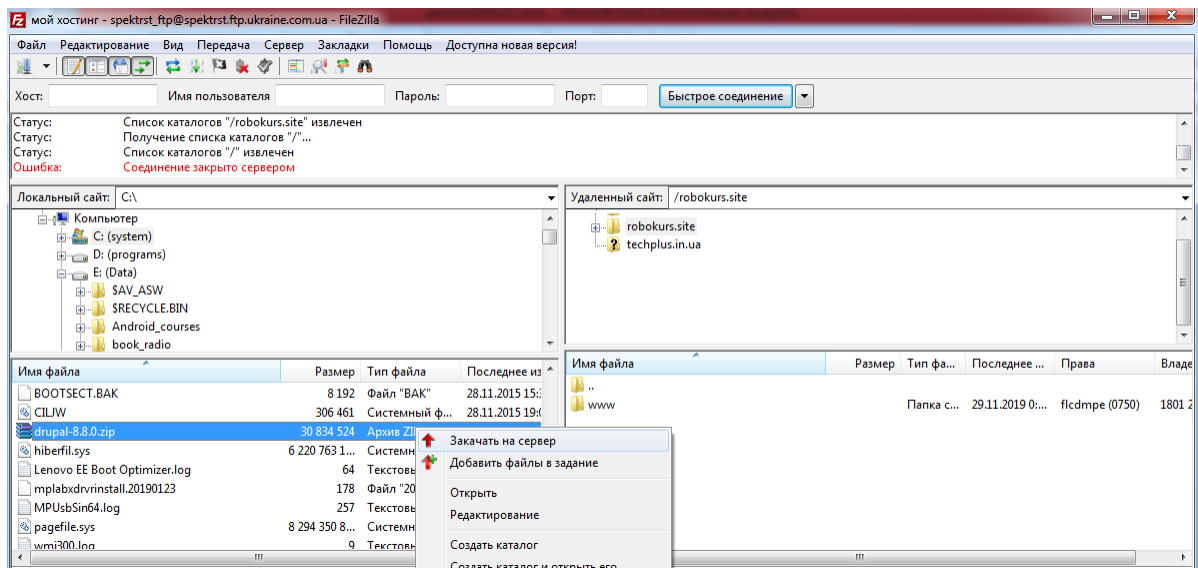


Рисунок 3.3 – Завантаження файлу з CMS Drupal на сервер хостингу за допомогою FTP менеджера FileZilla

г) розпакувати архів з інсталянтами при цьому буде створено новий каталог. Після цього видаляємо архів та змінимо ім'я каталогу на відповідне доменному імені;

д) встановити Drupal вказуючи базу даних в якій будуть зберігатися наші дані.

У другому випадку використовуємо панель хостингу, у нашому випадку на хостингу Ukraine.com.ua:

а) оскільки на хостингу перед встановленням будь якої CMS має бути вже створена – в першу чергу створюємо БД з ім'ям spektrst_robokur (рис. 3.4);

Создание базы данных ✕

Имя базы данных: 7/7

Кодировка:

Создать пользователя с таким же именем и полным доступом к базе данных

Имя базы данных может содержать только символы латиницы и цифры (максимум 16 символов).

Вы можете создать до 999 баз данных MySQL (уже создано 10, осталось 989)

Рисунок 3.4 – Створення порожньої БД на хостингу

б) обираємо пункт «Встановлення CMS», обираємо пункт з відповідним логотипом «Drupal»;

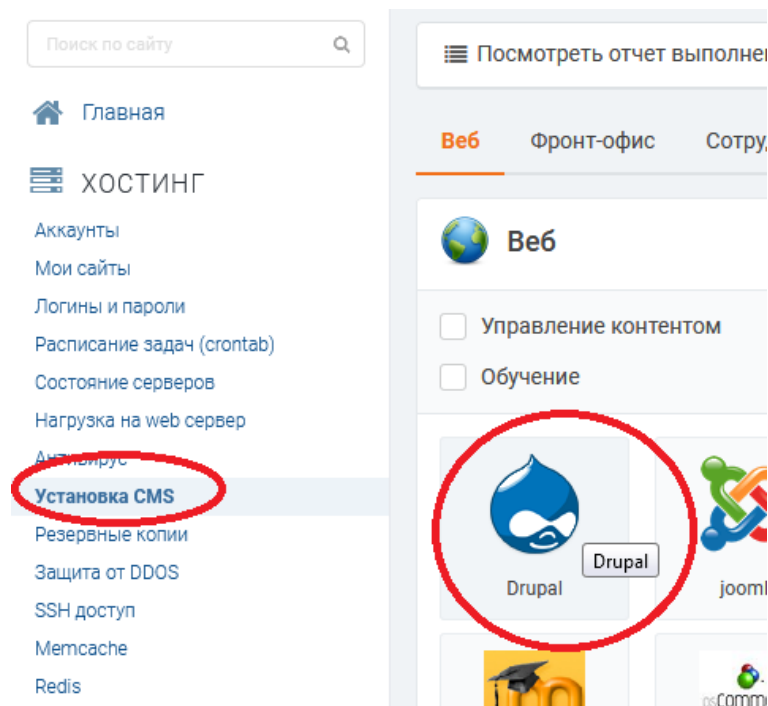


Рисунок 3.5 – Вибір CMS для встановлення

в) вказуємо параметри сайту:

- доменне ім'я;
- база даних;
- данні авторизації адміністратора сайту;

- електронна пошта адміністратора;
- назва сайту.

The image shows a web form for configuring a Drupal site. It has several input fields and a submit button. The fields are: 'Сайт:' with the value 'www.robokurs.site', 'База данных:' with 'spektrst_robokur', 'Administrator's login' with 'admin', 'Password' (empty), 'Administrator's email' with 'test@test.com', and 'Site name' with 'Drupal'. Below these is a checkbox labeled 'Я соглашаюсь с лицензионным соглашением' which is unchecked. At the bottom is a blue button labeled 'Установить'.

Рисунок 3.6 – Введення даних про сайт при встановленні CMS

Після установки і прив'язки порожня база даних має структуру приведену в файлі 0.xml

По суті при виклику локальної адреси для установки Друпал відбувається виконання на сервері Php коду який має наступний вигляд:

```
<? Php
```

```
define ( 'DRUPAL_ROOT', getcwd ()); – отримуємо ім'я поточного
робочого каталогу і присвоюємо його в якості значення константи
DRUPAL_ROOT тобто по суті кореневого каталогу Друпал
```

```
require_once DRUPAL_ROOT. '/Includes/bootstrap.inc'; – включаємо і
виконуємо файл bootstrap.inc в якому задаються константи і функції
(bootstrap перекладається як початкова завантаження). Файл bootstrap.inc з
російськими коментарями представлений в папці зі зміненими інсталянтами.
```

```
drupal_bootstrap (DRUPAL_BOOTSTRAP_FULL); – виклик функції з
підключеного файлу
```

```
menu_execute_active_handler ();
```

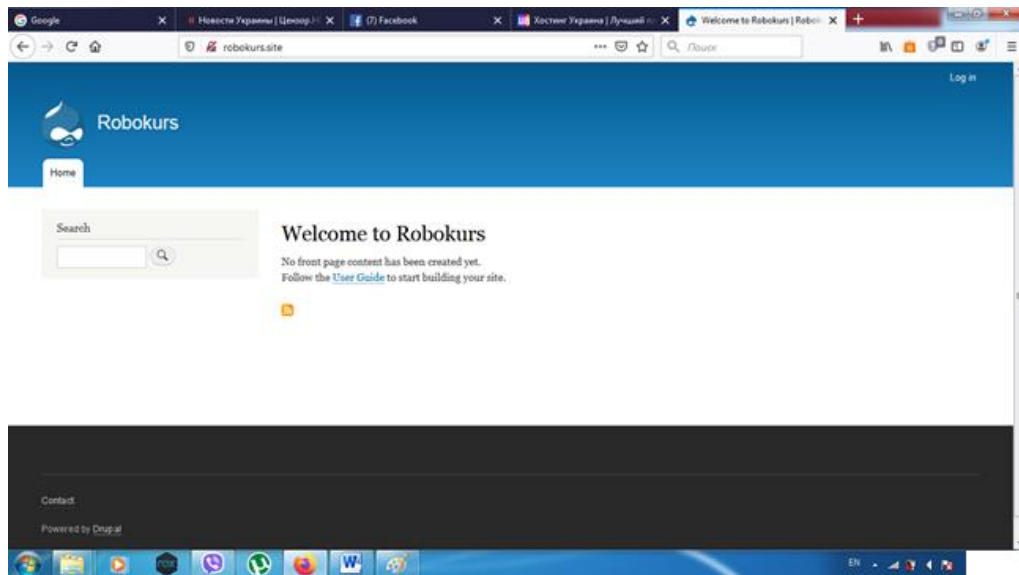


Рисунок 3.7 – Зовнішній вигляд сайту після встановлення CMS

Після того як CMS встановлено ми отримуємо базову структуру сайту. Наступним етапом є створення зовнішнього вигляду кінцевого сайту, що включає в себе наступні операції:

- вибір мови користувача;
- розробка назви та слогана (гасла);
- розробка логотипу;
- розробка зовнішнього оформлення сайту;
- розробка меню.

Для роботи з різними мовами користувача CMS Drupal має власні готові інструменти. Для їх підключення необхідно увійти на сайт під адміністраторським паролем, обрати пункт розширення та встановити галочки напроти компонентів в області «Multilingual» як показано на рис. 3.8.

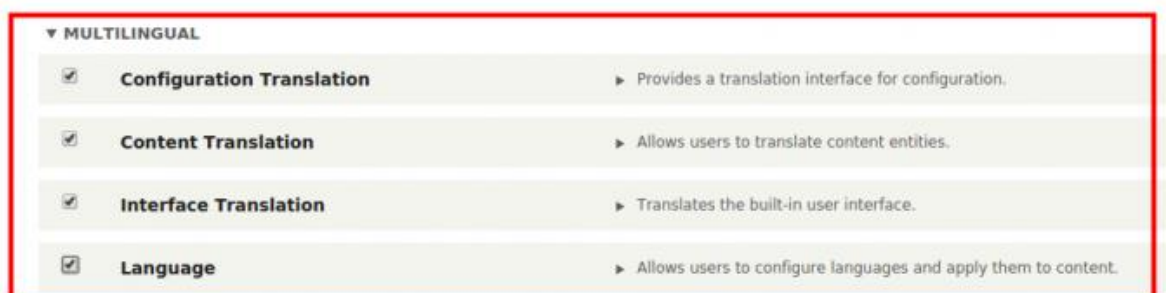


Рисунок 3.8 – Активація модулів для роботи з різними мовами оформлення

Наступним кроком для активації мов користувача є встановлення додаткових мов в меню «Конфігурація» сайту в розділі «Мова та переклад матеріалу» (рис. 3.9).

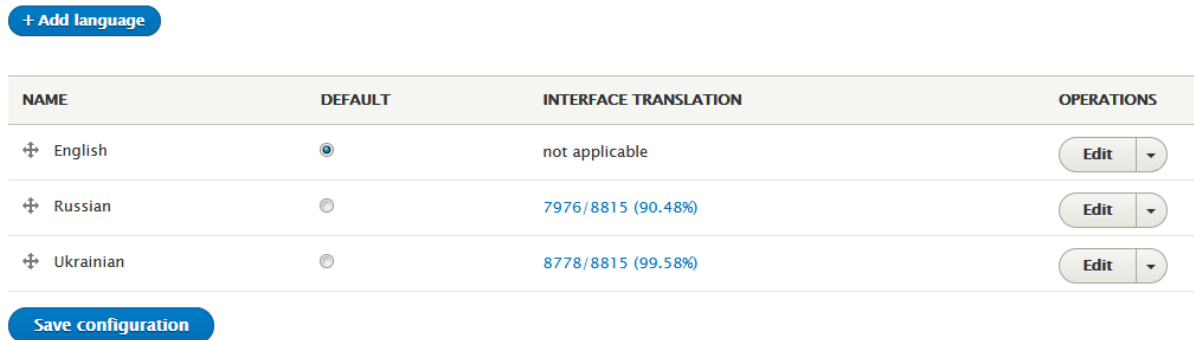


Рисунок 3.9 – Встановлення та підключення російської та української мови для оформлення сайту

Після установки та вибору у якості мови за замолучанням української мови всі адміністративні пункти меню автоматично будуть переведені.

Наступним кроком є зміна назви сайту, що відображається в заголовку закладки браузера та в верхній частині сайту в залежності від його оформлення. Також це основний тег сайту. Для зміни назви та гасла сайту обираємо пункт «Основні налаштування сайту» в пункті адміністративного меню «Конфігурація» та вводимо у нашому випадку назву «Інформатика 7 клас» та гасло «Інтерактивний навчальний курс з елементами робототехніки» (рис. 3.10).

ДЕТАЛІ САЙТУ

Назва сайту *

Гасло

Короткий вислів, що характеризує ваш сайт.

Рисунок 3.10 – Зміна назви та гасла сайту

Оскільки курс, що нами розробляється, є авторським та унікальним – створення логотипу зробимо також самостійно та не будемо завантажувати готовий з інтернету.

Для цього були обрані базові малюнки, що характеризують направленість роботи сайту. Наш логотип має поєднувати знання, які буде характеризувати книга, учні отримують знання за допомогою комп'ютера, отже він також має бути присутнім та ми вивчаємо мову C++ з використання платформи Arduino, а відповідно треба використати відповідні значки. Для створення логотипу сайту було обрано графічний редактор Photoshop (рис. 3.11).

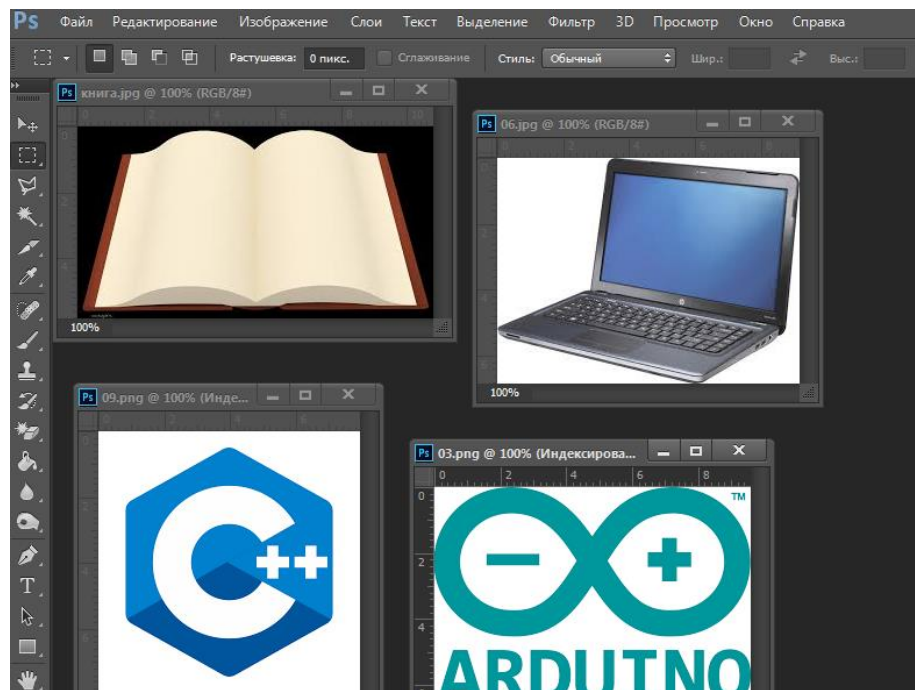


Рисунок 3.11 – Завантаження малюнків для створення логотипу

Значки мови програмування C++ та платформи Arduino були розміщені таким чином, щоб вони знаходились всередині монітора комп'ютера, комп'ютер було поміщено на книгу та було додано цифру «7», що характеризує 7 клас. Отриманий рисунок для логотипу наведено на рис. 3.12 та збережено у форматі PNG оскільки цей формат дозволяє залишати прозорим фон зображення.

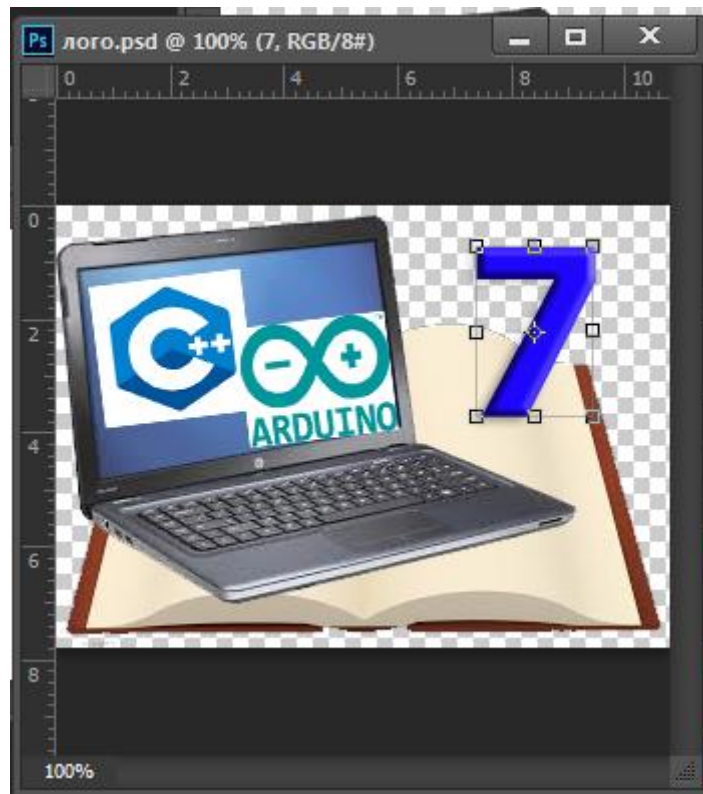


Рисунок 3.12 – Кінцевий результат створення логотипу у програмі Photoshop

Крім створення логотипу його треба додатково зберігти у формати значка ісо, щоб він відображався зверху у закладинці браузера. Для цього використовуємо будь який онлайн сервіс та зберігаємо значок з розширенням 32x32 точок.

Наступним етапом є створення зовнішнього оформлення сайту, тобто шаблонів оформлення сторінок різних типів та блоків сайту.

CMS Drupal має власні інструменти для створення шаблонів сторінок та блоків.

Головним файлом теми є файл info.yml, який задає всю базову інформацію. Це замена.info файлів сьомого Друпала, їх використання дає більше можливостей і гнучкості в налаштуванні теми. Аналогічні файли створюються для модулів, тому важливо задати для ключа «type» значення «theme». Під тема створюється, як і будь-яка тема, тільки додається сертифікат «base theme».

Name (обов'язково) – читабельним для людей (human-readable) назва теми, яке буде відображатися в списку тем на сторінці Appearance

description (обов'язково) – опис, який буде так само висвітлюватися на сторінці Appearance;

package – назва групи, в яку групуються теми з таким значення цього ключа;

type (обов'язково) – тип розширення, для тих завжди буде мати значення «theme»;

base theme – базова (батьківська) тема для даної, якщо ми створюємо під тему

core (обов'язково) – версія Друпала, з якої наша тема сумісна;

version – версія модуля, якщо він розміщується на drupal.org;

screenshot – фотографія (може бути скріншот), який буде відображатися на сторінці Appearance. Якщо не поставити цього ключа, тоді Drupal шукати файл з ім'ям «screenshot.png» в папці з темою, щоб відобразити;

libraries – бібліотеки, що містять css і js файли, які будуть додаватися до всіх сторінок. Про додаванні бібліотек є ціла стаття на drupal.org. Якщо коротко, то до теми додається ще один файл типу my_theme.libraries.yml

Breakpoints – це точки, в яких сайт змінює розташування елементів, в залежності від ширини екрану, використовуючи media query в CSS. Брейкпоінта задаються в файле.breakpoints.yml. Після установки їх в темі, до них будуть мати доступ різні модулі або інші теми, які створюють функціональність, залежить від брейкпоинтов.

Drupal 8 користується Twig. Це двигун шаблонів (template engine) для PHP і є частиною фреймворка Symfony. Всі функції типу theme_* і файли *.tpl.php замінені twig файлами *.html.twig.

Drupal 8 дозволяє перезаписувати (overriding) будь-які шаблони, які використовуються для генерації html-розмітки. Якщо потрібно перезаписати шаблон ядра чи іншої теми, потрібно в директорію templates, що лежить в директорії теми додати *.html.twig файл з відповідним ім'ям.

Будемо створювати тему з ім'ям robokurs.

Створимо на сервері в директорії: `"/ themes / custom"` папку robokurs для нашої теми. І створимо в ній файл `robokurs.info.yml` з описом теми, такого змісту:

```
# Название темы.
name: Robokurs
# Это тема, а не модуль и тому подібне
type: theme
description: 'Theme for Robokurs.'
# version: VERSION
# core: 8.x
libraries:
  - robokurs/global-styling
ckeditor_stylesheets:
  - css/ckeditor-iframe.css
# Regions
# Footer listed first to make it the default region.
regions:
  footer: 'Footer'
  header: 'Header'
  help: 'Help'
  menubar: 'Menu bar'
  highlighted: 'Highlighted'
  banner_top: 'Banner top'
  top_column_first: 'Top first column'
  top_column_second: 'Top second column'
  top_column_third: 'Top third column'
  top_column_fourth: 'Top fourth column'
  breadcrumb: Breadcrumb
  content: 'Content'
```

```

sidebar_first: 'Sidebar first'
sidebar_second: 'Sidebar second'
bottom_column_first: 'Bottom first column'
bottom_column_second: 'Bottom second column'
bottom_column_third: 'Bottom third column'
bottom_column_fourth: 'Bottom fourth column'
banner_bottom: 'Banner bottom'
footer_column_first: 'Footer first column'
footer_column_second: 'Footer second column'
footer_column_third: 'Footer third column'
footer_column_fourth: 'Footer fourth column'

```

```
#plugins
```

```
plugins:
```

```
  page_layout:
```

```
    – layouts: 'layouts/core'
```

```
version: '8.x-1.3'
```

```
core: '8.x'
```

```
project: 'Robokurs'
```

```
datestamp: 1462820623
```

Помістимо в папку теми її скріншот (скріншот макета верстки) для відображення на сторінці включення тим, з вказаним ім'ям (screenshot.png). Оскільки тема ще в процесі розробки у якості цього файлу помістимо наш розроблений логотип.

Створимо в папці теми файл robokurs.breakpoints.yml з описом "Media Queries" для модуля: Breakpoint. Надалі при використанні залежних від Breakpoint модулів, наприклад: Responsive image, можна використовувати задані в темі "Media Queries".

```
robokurs.smalltouch:
```

```
  label: smalltouch portrait
```

```
  mediaQuery: 'only screen and(min-width: 0px) and (max-width: 320px)'
```

weight: 4

multipliers:

– 1x

robokurs.smalltouch_landscape:

label: smalltouch landscape

mediaQuery: 'only screen and (min-width: 321px) and (max-width: 480px)'

weight: 3

multipliers:

– 1x

robokurs.tablet:

label: tablet portrait

mediaQuery: 'only screen and (min-width: 481px) and (max-width: 768px)'

weight: 2

multipliers:

– 1x

robokurs.tablet_landscape:

label: tablet landscape

mediaQuery: 'only screen and (min-width: 769px) and (max-width: 1024px)'

weight: 1

multipliers:

– 1x

robokurs.big_screen:

label: big screen

mediaQuery: 'only screen and (min-width: 1025px)'

weight: 0

multipliers:

– 1x

Опишемо підключення CSS і JS бібліотек до теми на прикладі підключення CSS для власних стилів теми і Bootstrap, для це створимо файл `robokurs.libraries.yml` в папці теми, такого змісту:

```
global-styling:
  version: VERSION
  css:
    theme:
      css/style.css: {}
      css/colors.css: {}
color.preview:
  version: VERSION
  css:
    theme:
      color/preview.css: {}
      css/admin-layout-settings.css: {}
  js:
    color/preview.js: {}
    js/robokurs.js: {}
  dependencies:
    - color/drupal.color
# - core/drupalSettings
maintenance-page:
  version: VERSION
  css:
    theme:
      css/maintenance-page.css: {}
      css/colors.css: {}
  dependencies:
    - system/maintenance
    - robokurs /global-styling
```

robokurs-columns:

version: VERSION

js:

js/robokurs-columns.js: {}

dependencies:

- core/jquery
- core/drupal
- core/drupalSettings

responsive-layout:

version: VERSION

css:

theme:

public://mayo/mayo_files/robokurs.responsive.layout.css: {}

dependencies:

- mayo/global-styling

black-menu:

version: VERSION

css:

theme:

css/black-menu.css: {}

dependencies:

- robokurs/global-styling

superfish:

version: VERSION

css:

theme:

css/mayo-superfish.css: {}

dependencies:

- mayo/global-styling

fontsize:

```

version: VERSION
js:
  js/mayo-fontsize.js: {}
dependencies:
  - core/jquery
messages:
  version: VERSION
  css:
    component:
      css/messages.css: { preprocess: false }

```

У файлі `robokurs.theme` (в папці теми) повинні міститися `preprocess`–функції використовуються в створюваній темі. Так як потреба в них поки відсутня, що не будемо створювати даний файл.

В директорії `css` знаходяться файли стилів.

В директорії `js` – файли для javascript або викликів Функ бібліотеки JQuery.

В директорії `images` фонові картинки.

В директорії `templates`– шаблони відображення нод і сторінок.

Отже після створення шаблону згідно з прописаними даними ми отримали наступні блоки:

- `footer: 'Footer'` – нижній горизонтальний блок;
- `header: 'Header'` – верхній горизонтальний блок;
- `help: 'Help'` – блок з текстом допомоги користувачеві по

користуванням сайтом;

- `menubar: 'Menu bar'`;
- `highlighted: 'Highlighted'`;
- `banner_top: 'Banner top'`;
- `top_column_first: 'Top first column'`;
- `top_column_second: 'Top second column'`;
- `top_column_third: 'Top third column'`;

- top_column_fourth: 'Top fourth column';
- breadcrumb: Breadcrumb – так звані «Хлібні крихти» – шлях відносно основного блоку сайту згідно пунктів меню ;
- content: 'Content' – блок зі змістом сторінок;
- sidebar_first: 'Sidebar first' – перший блок лівого меню;
- sidebar_second: 'Sidebar second' – другий блок лівого меню;
- bottom_column_first: 'Bottom first column' – перший нижній прямокутний блок з додатковим змістом, що розташовується над блоком 'Footer';
- bottom_column_second: 'Bottom second column' – другий нижній прямокутний блок з додатковим змістом;
- bottom_column_third: 'Bottom third column' – третій нижній прямокутний блок з додатковим змістом;
- bottom_column_fourth: 'Bottom fourth column' – четвертий нижній прямокутний блок з додатковим змістом;
- banner_bottom: 'Banner bottom';
- footer_column_first: 'Footer first column';
- footer_column_second: 'Footer second column';
- footer_column_third: 'Footer third column';
- footer_column_fourth: 'Footer fourth column';

Встановлюємо створену тему. Для цього переходимо на вкладку «Вигляд», натискаємо кнопку «+Встановити нову тему» та вказуємо папку з розробленою темою. Після цього вмикаємо її та робимо її базовою (рис. 3.13)

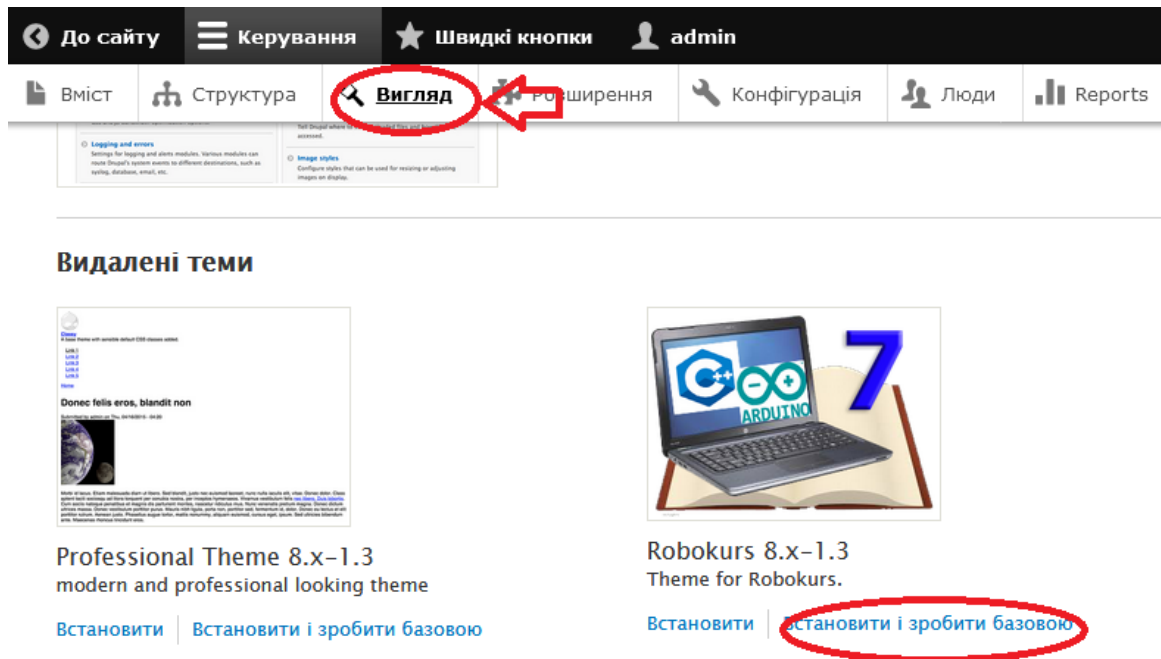


Рисунок 3.13 – Встановлення розробленої теми в якості базової

Наступним етапом проектування оформлення є розташування інформації в блоках. Для цього використовуємо меню «Структура»–«Схема блоків». У відповідних стовпчиках обираємо де буде розміщувати той чи інший зміст (рис. 3.14).

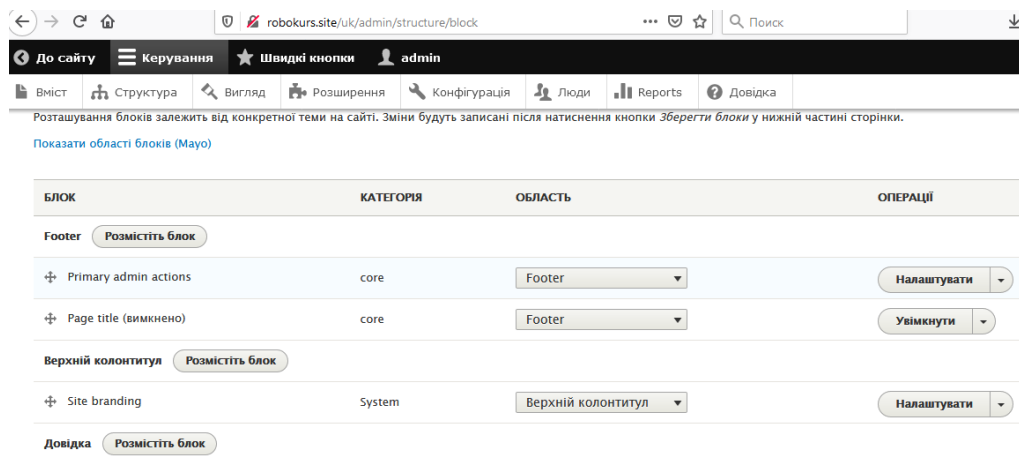


Рисунок 3.14 – Налаштування блоків

Наступним етапом є створення меню сайту. Пропонуємо створити 3 меню для керування та навігації по сайту:

– меню адміністратора для керування матеріалами та допусками користувачів;

- меню вчителя для роботи вчителя з сайтом;
- меню учня для перегляду навчальних матеріалів та проходження тестів.

Меню адміністратора має наступні пункти:

- додати новий матеріал;
- додати новий тип матеріалів;
- керування користувачами.

Меню вчителя має наступні пункти:

- керування навчальними матеріалами;
- мої учні;
- звернення учнів;
- налаштування зворотнього зв'язку учнів.

Меню учня має наступні пункти:

- головна;
- навчальні матеріали за темами;
- тести для перевірки отриманих знань;
- середовище для тестування написаної програми;
- питання, що найчастіше виникають;
- зв'язок з вчителем;
- моя статистика.

Для створення меню в пункті «Структура» обираємо пункт «Меню»

(рис. 3.15).

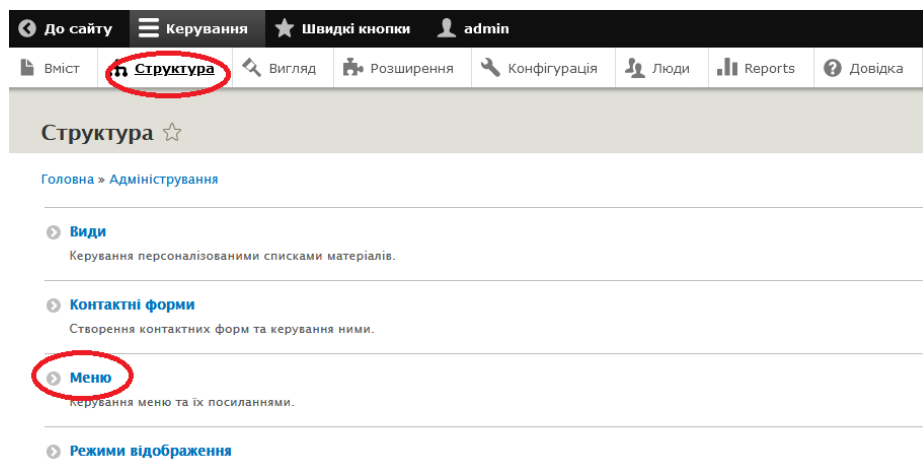


Рисунок 3.15 – Створення меню

На сторінці, що з'являється обираємо пункт «+Додати меню» та вводимо назву, адресу посилання та опис нового меню як наведено на рис. 3.16 для меню адміністратора.

Додати меню ☆

Головна » Адміністрування » Структура » Меню

You can enable the newly-created block for this menu on the [Block layout page](#).

Title *

Меню адміністратора

Назва меню *

Adminmenu

Унікальна назва для побудови URL-адреси меню. Вона має містити лише латинські літери нижнього регістру, цифри та дефіси

Адміністративна інформація

атора для керування матеріалами та допусками користувачів

Мова меню

Ukrainian

Зберегти

Рисунок 3.16 – Вказання параметрів меню на прикладі «Меню адміністратора»

Далі для кожного меню робимо пункти меню обравши «Додати посилання» в області «Посилання меню» та вводимо параметри нового пункту меню.

Назва посилання меню *

Керування користувачами

Текст, що буде використаний для цього посилання в меню

Посилання *

/uk/admin/people

- Місце, на яке вказує це посилання меню.
- Почніть вводити частину назву матеріалу для того, щоб обрати його. Також може ввести внутрішній шлях матеріалу, як-от `/node/add`, або зовнішню URL-адресу на кшталт `http://example.com`. Для посилання на головну сторінку введіть `<front>`.

Увімкнено

Прапор, що вказує, чи має це посилання бути увімкненим в меню, чи прихованим.

Опис

Додавання, видалення або зміна параметрів для користувачі

Показуване при наведенні мишею на посилання меню.

Show as expanded

Якщо позначено і це меню має дочірні меню, воно завжди буде розкритим.

Батьківське посилання

<Меню адміністратора>

Максимальна глибина посилання та усіх його нащадків зафіксована. Деякі посилання меню можуть бути недоступні як батьки, якщо їх вибір буде перевищити це обмеження.

Вага

0

Рисунок 3.17 – Створення нового посилання в меню на прикладі «Меню адміністратора»

Після створення всіх трьох додаткових меню вказуємо їх розташування в пункті «Структура» – «Схема блоків» у першому, другому та третьому боковому лівому блоці відповідно (рис. 3.18).

Вміст		Розмістіть блок			
Main page content	System	Вміст	1	Налаштувати	
Sidebar first		Розмістіть блок			
Знайти	Форми	Sidebar first	-1	Налаштувати	
Меню адміністратора	Меню	Sidebar first	0	Налаштувати	
User account menu	Меню	Sidebar first	11	Налаштувати	
Main navigation (вимкнено)	Меню	Sidebar first	10	Увімкнути	

Рисунок 3.18 – Розміщення меню в блоках сайту

3.2 Заповнення навчальним контентом

Останнім етапом оформлення сайту є розміщення контенту. Для цього переходимо до меню «Вміст» натискаємо кнопку «+ Додати вміст» (рис. 3.19).

Вміст

Вміст ☆

Вміст Коментарі Файли

Головна » Адміністрування

+ Додати вміст

Заголовок Тип матеріалів Стан публікації Мова

- Усі - - Усі - - Усі -

Фільтр

ЗАГОЛОВОК ТИП МАТЕРІАЛІВ АВТОР СТАН ОНОВ

Рисунок 3.19 – Додавання вмісту

Оскільки нам необхідно буде працювати з скриптами, що будуть написані самостійно на php та вставляти на сторінки за допомогою iframe – нам додатково треба встановити модулі, що додають текстовий формат «PHP» та дозволять розміщувати власний PHP код не зв'язаний з ядром CMS Drupal.

Методологічною основою навчального курсу, що складається з двох частин було обрано:

– для першої частини, що включає загальні знання в області інформатики – підручник «Інформатика 7 клас» авторів Й.Я.Ривкінд, Т.І.Лисенко, Л.А.Чернікова, В.В.Шакотько київського видавництва, що рекомендовано Міністерством освіти та науки України та підручник «Інформатика» авторів А.М. Гуржій, Л.А. Карташова, В.В. Лапінський, В.Д. Руденко львівського видавництва також рекомендованого Міністерством освіти та науки України;

– для другої частини, що включає базові знання в області програмування на мові C++ платформи Arduino з різних джерел обрано найпростіші, однак при цьому найцікавіші завдання.

Загалом перелік тем для навчального курсу має наступний вигляд:

Тема 1 – Електронне листування

Урок 1.1 – Електронна поштова скринька

Урок 1.2 – Додавання файлів до листів. Перенаправлення пошти

Тема 2 – Моделювання

Урок 2.1 – Типи моделей. Форми подання інформаційної моделі

Урок 2.2 – Побудова інформаційних моделей

Тема 3 – Алгоритми

Урок 3.1 – Категорії команд Скретч

Урок 3.2 – Найпростіші алгоритми з повторенням

Урок 3.3 – Типи даних середовища Скретч

Урок 3.4 – Константи, змінні й списки

Урок 3.5 – Алгоритми з розгалуженням

Урок 3.6 – Алгоритми підвищеної складності

Тема 4 – Обробка даних в таблицях

Урок 4.1 – Електронні таблиці, табличний процесор

Урок 4.2 – Дані в електронних таблицях

Урок 4.3 – Формули в електронних таблицях

Урок 4.4 – Вбудовані функції

Урок 4.5 – Діаграми

Урок 4.6 – Моделювання в електронних таблицях

Тема 5 – Основи роботи з Arduino

Урок 5.1 – Що таке Arduino та що з нього можна зробити

Урок 5.2 – Середовище розробки програм для платформи Arduino

Тема 6 – Основні блоки програм на мові C++ для Arduino

Урок 6.1 – Методи setup та loop

Урок 6.2 – Типи даних в C++

Урок 6.3 – Директива #define та її використання

Тема 7 – Перші кроки в програмуванні на C++ для Arduino

Урок 7.1 – Оператори присвоєння, порівняння

Урок 7.2 – Найпростіші апаратні засоби для Arduino

Урок 7.3 – Керуємо світлодіодом

Урок 7.4 – Світлодіод та кнопка

Тема 8 – Наш перший прилад на Arduino

Урок 8.1 – Світлофор, що керується кнопками

Урок 8.2 – Автоматичний світлофор

Тема 9 – Різні прилади індикації

Урок 9.1 – Рідинно-кристалічний текстовий індикатор;

Урок 9.2 – Світлодіодна матриця

Урок 9.3 – Графічний дисплей

Тема 10 – Робота з простими датчиками

Урок 10.1 – Фотодіод

Урок 10.2 – Датчик температури та вологості

Урок 10.3 – Ультразвуковий датчик

Для спрощення редагування матеріалів та можливості додавання мультимедійних матеріалів, таких як фото, аудіо, відеоматеріали встановлюємо редактор CKEditor Media Embed Plugin у якості додаткового функціонала.

Курс має можливість додавання відеоматеріали, що знаходяться на сторонньому сервісі – наприклад на youtube. Це як правило використовується для того, щоб не займати багато місця на сервері сайту.

3.3 Робота з додатковими та основними таблицями бази даних та створення додаткового функціоналу

У другому розділі було описано структуру додаткових таблиць бази даних для оцінювання рівня засвоєння матеріалів учнями курсу та подальшою розробки програмної автоматичної перевірки знань учнів. Однак ці таблиці треба ще створити.

Для створення нових таблиць бази даних скористаємось вбудованою в хостинг утилітою phpMyAdmin (рис. 3.20) та будемо їх створювати за допомогою SQL запитів.

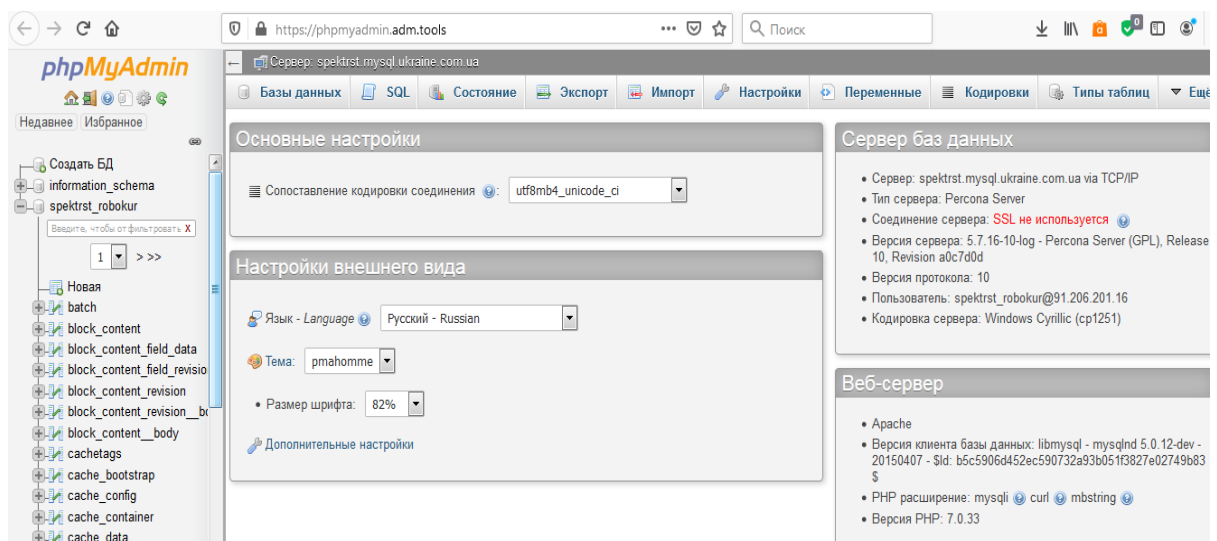


Рисунок 3.20 – Використання інструменту phpMyAdmin

Створюємо SQL запити для створення бази даних.

Для створення таблиці Autors запит буде мати наступний вигляд:

```
CREATE TABLE Autors (id INT, _login VARCHAR(50),
password_md5hash VARBINARY(MAX), facebook_acc VARCHAR(MAX),
access_level INT, first_name VARCHAR(50), last_name VARCHAR(50),
PRIMARY KEY (id));
```

Для створення таблиці Users запит буде мати наступний вигляд:

```
CREATE TABLE Users (id INT, _login VARCHAR(50),
password_md5hash VARBINARY(MAX), facebook_acc VARCHAR(MAX),
access_level INT, first_name VARCHAR(50), last_name VARCHAR(50),
idAutor INT, PRIMARY KEY (id), FOREIGN KEY (idAutor) REFERENCES
Autors (id));
```

Для створення таблиці Questionnaires запит буде мати наступний вигляд:

```
CREATE TABLE Questionnaires (id INT, Name VARCHAR(250), idAutor
INT, Create_Date DATE, QCount INT, PRIMARY KEY (id), FOREIGN KEY
(idAutor) REFERENCES Autors (id));
```

Для створення таблиці Res_Questionnaires запит буде мати наступний вигляд:

```
CREATE TABLE Res_Questionnaires (id INT, idQuestionnaires INT,
idUser INT, Create_Date SMALLDATETIME, GPS VARCHAR (MAX),
PRIMARY KEY (id), FOREIGN KEY (idQuestionnaires) REFERENCES
Questionnaires (id), FOREIGN KEY (idUser) REFERENCES Users (id));
```

Для створення таблиці Texts запит буде мати наступний вигляд:

```
CREATE TABLE Texts (id INT, idQuestionnaires INT, text_var TEXT,
PRIMARY KEY (id));
```

Для створення таблиці Questions запит буде мати наступний вигляд:

```
CREATE TABLE Questions (id INT, idQuestionnaires INT,
idQuestionnaire INT, Quest_type INT, Group_type INT, idText INT, Answ_count
```


INT, PRIMARY KEY (id), FOREIGN KEY (idQuestionnaire) REFERENCES Questionnaires (id), FOREIGN KEY (idText) REFERENCES Texts (id));

Для створення таблиці Res_Questions запит буде мати наступний вигляд:

```
CREATE TABLE Res_Questions (id INT, idRes_Questionnaire INT, idQuestion INT, PRIMARY KEY (id), FOREIGN KEY (idRes_Questionnaire) REFERENCES Res_Questionnaires (id), FOREIGN KEY (idQuestion) REFERENCES Questions (id));
```

Для створення таблиці Answers запит буде мати наступний вигляд:

```
CREATE TABLE Answers (id INT, idQuestion INT, answer_type INT, idText INT, PRIMARY KEY (id), FOREIGN KEY (idQuestion) REFERENCES Questions (id), FOREIGN KEY (idText) REFERENCES Texts (id));
```

Для створення таблиці ResTexts запит буде мати наступний вигляд:

```
CREATE TABLE ResTexts (id INT, text_var VARCHAR(MAX), PRIMARY KEY (id));
```

Для створення таблиці Res_Answers запит буде мати наступний вигляд:

```
CREATE TABLE Res_Answers (id INT, idRes_Question INT, idAnswer INT, idText INT, PRIMARY KEY (id), FOREIGN KEY (idRes_Question) REFERENCES Res_Questions (id), FOREIGN KEY (idAnswer) REFERENCES Answers (id), FOREIGN KEY (idText) REFERENCES ResTexts (id));
```

Для створення таблиць запитамі в phpMyAdmin переходимо у вкладку SQL та розміщуємо запити, після чого натискаємо кнопку «Вперед» (рис. 3.21).

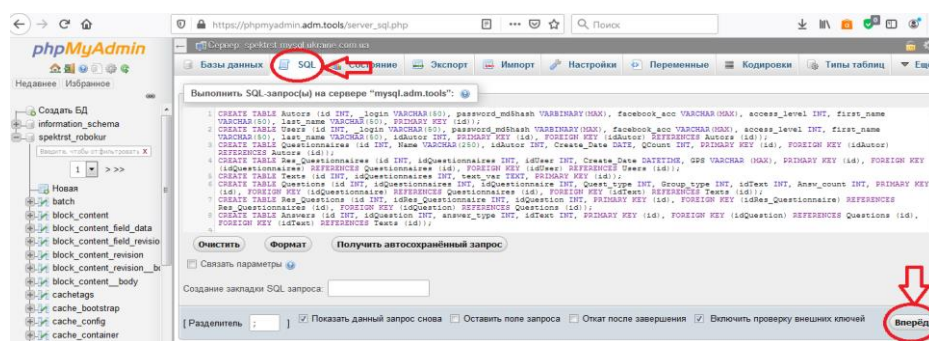


Рисунок 3.21 – Виконання запитів для створення додаткових таблиць БД в phpMyAdmin

Для роботи з даними, що зберігаються в базі даних були створені та використані наступні методи.

Для створення SELECT запитів використовуємо функцію db_select().

Простий запит виборки nid и title з таблиці node где nid >= 5:

```
//SQL запит
SELECT n.nid, n.title FROM node n WHERE n.nid >= 5

//php код
$query = db_select('node', 'n');
$query->fields('n', array('nid', 'title'));
$query->condition('n.nid', 5, '>=');
$result = $query->execute();
```

Отримання кількості записів в таблиці на прикладі node де nid > 10:

```
//SQL запит
SELECT COUNT(*) FROM node n WHERE n.nid > 10

//php код
$query = db_select('node', 'n');
$query->fields('n');
$query->condition('n.nid', 10, '>');
$result = $query->execute()->rowCount();
```

Запит з одним логічним оператором AND та оператором LIKE:

```
//SQL запит
SELECT n.nid, n.title FROM node n WHERE n.nid >= 5 AND n.title LIKE
'Chapter%'

//php код
$query = db_select('node', 'n');
$query->fields('n', array('nid', 'title'));
$query->condition('n.nid', 5, '>=');
$query->condition('n.title', 'Chapter%', 'LIKE');
$result = $query->execute();
```

Запит з використанням оператора IN:

```
//SQL запит
SELECT n.nid, n.title FROM node n WHERE n.nid IN (1, 5, 10, 11)

//php код
$in = array(1, 5, 10, 11);

$query = db_select('node', 'n');
$query->fields('n', array('nid', 'title'));
$query->condition('n.nid', $in, 'IN');
$result = $query->execute();
```

У випадку складних умов можна використовувати метод where як у наступному прикладі:

```
//SQL запит
SELECT n.title FROM node n
WHERE FROM_UNIXTIME(n.created) < CURDATE() - INTERVAL 1 DAY

//php код
$query = db_select('node', 'n');
$query->fields('n', array('title'));
$query->where('FROM_UNIXTIME(:created) < CURDATE() - INTERVAL 1 DAY',
array(':created' => $timestamp));
$result = $query->execute();
```

Запит з використанням функції SUM з групуванням:

```
//SQL запит
SELECT SUM(vv.value) AS vote_sum, vv.entity_id, n.title FROM votingapi_vote
vv
INNER JOIN node n ON n.nid = vv.entity_id
GROUP BY n.nid

//php код
$query = db_select('votingapi_vote', 'vv');
$query->addExpression('SUM(vv.value)', 'vote_sum');
$query->fields('vv', array('entity_id'));
$query->fields('n', array('title'));
$query->innerJoin('node', 'n', 'n.nid = vv.entity_id');
$query->groupBy('n.nid');
$result = $query->execute();
```

Запит з використанням підзапиту (subquery)

```
//SQL запит
SELECT n.title FROM node n
WHERE n.nid IN (SELECT ct.nid FROM custom_table ct)

//php код
$query = db_select('node', 'n');
$query->fields('n', array('title'));

$subquery = db_select('custom_table', 'ct');
$subquery->fields('ct', array('nid'));

$query->condition('n.nid', $subquery, 'IN');
$result = $query->execute();
```

Для додавання нових записів у БД використовуємо функцію `db_insert()`.

Додавання нового запису в таблицю node:

```
//SQL запит
INSERT INTO node (title, uid, type) VALUES ('New node', 1, 'article')

//php код
db_insert('node')
->fields(array(
    'title' => 'New node',
    'uid' => 1,
    'type' => 'article',
))
->execute();
```

Для оновлення записів в таблицях БД використовується функція `db_update()`.

```
//SQL запит
UPDATE node SET title = 'Updated', changed = 1363104629 WHERE nid = 1

//php код
$query = db_update('node');
$query->fields(array(
    'title' => 'Updated',
    'changed' => REQUEST_TIME,
));
$query->condition('nid', 1);
$num_updated = $query->execute();
```

```
//SQL запит
UPDATE node SET title = CONCAT(title, '(updated)') WHERE nid > 10

//php код
$query = db_update('node');
$query->expression('title', 'CONCAT(title, :suffix)', array(':suffix' => '(updated)'));
$query->condition('nid', 10, '>');
$num_updated = $query->execute();
```

Для видалення записів використовується функція `db_delete()`.

```
//SQL запит
DELETE FROM node WHERE nid > 10

//php код
$query = db_delete('node');
$query->condition('nid', 10, '>');
$num_deleted = $query->execute();
```

Одним з важливих особливостей навчального курсу є можливість швидкого зв'язку учня з вчителем за рахунок відправлення повідомлень на меседжери згідно з налаштуваннями зворотнього зв'язку зроблених вчителем («Меню вчителя» – «Налаштування зворотнього зв'язку учнів»).

Для цього було реалізовано 3 скрипти для відправлення повідомлень відповідно у Viber, Telegram та WhatsApp.

Розглянемо скрипт для відправлення повідомлень на Viber.

```
class Viber
{
    private $url_api = "https://chatapi.viber.com/pa/";

    private $token = "";

    public function message_post
    (
        $from,           // ID адміністратора сайту.
        array $sender,  // Данні відправителя.
        $text            // Текст.
    )
    {
        $data['from']    = $from;
        $data['sender']  = $sender;
        $data['type']    = 'text';
        $data['text']    = $text;
        return $this->call_api('post', $data);
    }

    private function call_api($method, $data)
    {
        $url = $this->url_api.$method;

        $options = array(
            'http' => array(
                'header' => "Content-type: application/x-www-form-
urlencoded\r\nX-Viber-Auth-Token: ".$this->token."\r\n",
                'method' => 'POST',
                'content' => json_encode($data)
            )
        );
```

```

        )
    );
    $context = stream_context_create($options);
    $response = file_get_contents($url, false, $context);
    return json_decode($response);
}
}
$Viber = new Viber();
$Viber->message_post(
    '01234567890A=',
    [
        'name' => 'Admin', // Имя отправителя. Максимум символов 28.
        'avatar' => 'http://robokurs.site/avatar.jpg' // Ссылка на аватарку.
    ],
    'Text'
);

```

Змінна \$token береться в Viber Admin Panel згідно документації <https://developers.viber.com/docs/api/rest-bot-api/#authentication-token>. Далі використовуючи створений клас створюється його екземпляр та методом message_post з отриманим токеном відправляється текст запитання вчителю на Viber у разі якщо в налаштуваннях вчителя цей тип відправлення повідомлень активований. Я якості аватарки відправителя використовуємо розроблений логотип сайту.

Розглянемо скрипт для відправлення повідомлень на Telegram.

Для початку нам необхідно створити нашого бота і отримати ключ. Для цих цілей розробниками створено бот "@BotFather". Нам необхідно написати боту в чат кілька повідомлень з потрібними нам параметрами:

а) для початку пишемо повідомлення `"/ start"`. нам у відповідь повернуть всі можливі команди бота;

б) відправляємо боту команду `"/ newbot"`;

в) нам пропонують ввести ім'я нашого бота. тут як вам буде завгодно, я назвав `"pechnectest"`;

г) тепер нам необхідно вибрати ім'я користувача для бота, головне, що б назва нашого бота закінчувалося на `"bot"` (наприклад `testbot` або `test_bot`). у моєму випадку `"pechnectest_bot"`;

г) після всього ми побачимо привітання і наш токен;

д) тепер додаємо в друзі нашого бота. можна шукати в назві що ми задали раніше.

Отримуємо Chat ID Telegram Bot –а.

Для того, що б нам відправляти повідомлення, нам необхідно дізнатися "chat id", якщо по–простому, то ідентифікатор нашого чату. Для цього нам необхідно виконати ряд дій:

а) написати щось нашому боту, абсолютно будь–яке повідомлення. Потрібно це для того, що б ми могли дістати наш chat id. Без цього метод API Telegram поверне нам порожнє значення.

б) в браузері виконуємо наступну команду:

`https://api.telegram.org/bot<token>/getUpdates`

<Token> – сюди вставляємо раніше отриманий токен.

в) в результаті ми отримаємо відповідь у форматі json в якому буде наш telegram чат ID.

Безпосередньо сам скрипт після реєстрації бота:

```
<?php
// тут токен бота
define('TELEGRAM_TOKEN', '999999999:XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX');

// тут внутрішній айдишник
define('TELEGRAM_CHATID', '999999999');

message_to_telegram('Привет!');

function message_to_telegram($text)
{
    $ch = curl_init();
    curl_setopt_array(
        $ch,
        array(
            CURLOPT_URL => 'https://api.telegram.org/bot' . TELEGRAM_TOKEN .
            '/sendMessage',
            CURLOPT_POST => TRUE,
            CURLOPT_RETURNTRANSFER => TRUE,
            CURLOPT_TIMEOUT => 10,
            CURLOPT_POSTFIELDS => array(
                'chat_id' => TELEGRAM_CHATID,
                'text' => $text,
            ),
        )
    );
    curl_exec($ch);
}
```

Коли це все реалізовано достатньо з форми зворотнього зв'язку виконати метод `message_to_telegram`

Розглянемо скрипт для відправлення повідомлень на WhatsApp.

```
$data = [
```

```

        'phone' => '380952478394', // Телефон отримувача
        'body' => 'До вас виникло питання', // Повідомлення
    ];
    $json = json_encode($data); // Закодуємо данні в JSON
    // URL для запиту POST /message
    $url = 'https://foo.chat-api.com/message?token=83763g87x';
    // Сформуємо контекст звичайного POST-запиту
    $options = stream_context_create(['http' => [
        'method' => 'POST',
        'header' => 'Content-type: application/json',
        'content' => $json
    ]
    ]);
    // Відправимо запит
    $result = file_get_contents($url, false, $options);

```

3.4 Реалізація онлайн опитування рівня знань учнів

Для реалізації можливості створення тестів з перевірки знань учнів було розроблено скрипт, що заповнює додаткові створені нами таблиці бази даних NewAnkete.php.

Оскільки перед дописуванням нових рядків в таблиці з тестами нам треба знати їх поточне максимальне значення початково в скрипті створені змінні:

```
$MaxAnketeIndex=0;//максимальний індекс анкети в БД
```

```
$MaxTextIndex=0;//максимальний індекс тексту в БД
```

```
$MaxQuestionIndex=0;//максимальний індекс питання в БД
```

```
$MaxAnswerIndex=0;//максимальний індекс відповіді в БД
```

Далі з'єднуємося з базою даних. Процедура з'єднання з БД має вигляд:

```

$dbcnx = @mysql_connect($dblocation,$dbuser,$dbpasswd);
if (!$dbcnx) // Если дескриптор равен 0 соединение не установлено
{
    echo("ERROR_DB_CONNECT_0<br>");
    exit();
}
if ($dbcnx>0)
{
    //echo("DB_CONNECT_OK<br>");
}
// Код з'єднання з БД
if (!@mysql_select_db($dbname, $dbcnx))
{

```

```

echo( "ERROR_DB_CONNECT_1<br>" );
exit();
}

```

Для визначення максимальних індексів в таблицях для подальшого створення нових записів було створено наступні функції.

Визначаємо максимальний індекс анкет в БД:

```

function GetMaxAnketeIndex($mysqli){
    $query="SELECT MAX(id) as _maxID FROM questionnaires;";
    $res = mysql_query($query);
    if ($res){
        while($row = mysql_fetch_array($res)) {
            $result=$row['_maxID'];
        }
    }
    return $result;
}

```

Визначаємо максимальний індекс тексту в БД

```

function GetMaxTextIndex($mysqli){
    $query="SELECT MAX(id) as _maxID FROM texts;";
    $res = mysql_query($query);
    if ($res){
        while($row = mysql_fetch_array($res)) {
            $result=$row['_maxID'];
        }
    }
    return $result;
}

```

Визначаємо максимальний індекс питання в БД

```

function GetMaxQuestionIndex($mysqli){
    $query="SELECT MAX(id) as _maxID FROM questions;";
    $res = mysql_query($query);
    if ($res){
        while($row = mysql_fetch_array($res)) {
            $result=$row['_maxID'];
        }
    }
    return $result;
}

```

Визначаємо максимальний індекс відповіді в БД:

```

function GetMaxAnswerIndex($mysqli){
    $query="SELECT MAX(id) as _maxID FROM answers;";
    $res = mysql_query($query);

```



```

    if ($res){
        while($row = mysql_fetch_array($res)) {
            $result=$row['_maxID'];
        }
    }
    return $result;
}

```

Для додавання записів в БД було також створено спеціальні функції

Додаємо новий текст з індексом \$textid в БД:

```

function AddNewText($mysqli,$textid, $newtext){
    $query="INSERT INTO texts VALUES (".$textid.", " ".$newtext.");";
    $res = mysql_query($query);
}

```

Додаємо нову анкету з індексом \$anketeid в БД:

```

function AddNewAnkete($mysqli,$anketeid, $name, $author, $respondents){
    $datenow=date('Y-m-d', time());
    $query="INSERT INTO questionnaires VALUES (".$anketeid.", " ".$name.",
    ".$author.", " ".$datenow.", " ".$respondents.");";
    $res = mysql_query($query);
}

```

Додаємо нове питання з індексом \$quest_id в БД:

```

function AddNewQuestion($mysqli,$quest_id, $ankete_id, $qtext_id, $answers_lim){
    $query="INSERT INTO questions VALUES( ".$quest_id.", " ".$ankete_id.", 0, 0,
    ".$qtext_id.", " ".$answers_lim.");";
    $res = mysql_query($query);
}

```

Додаємо нову відповідь з індексом \$answ_id в БД:

```

function AddNewAnswer($mysqli,$answ_id, $quest_id, $answ_type, $qtext_id){
    $query="INSERT INTO answers VALUES( ".$answ_id.", " ".$quest_id.",
    ".$answ_type.", " ".$qtext_id.");";
    $res = mysql_query($query);
}

```

Оскільки тексти питань та відповідей у БД у нас проіндексовані було створено функції перевірки наявності текста в БД, щоб не дублювати текст, а вказувати лише його індекс та функцію пошуку індексу тексту якщо він існує в БД.

Перевіряємо наявність тексту \$testtext в БД – функція повертає скільки раз текст зустрічається в БД:

```
function TestTextAvailability($mysqli, $testtext){
    $query="SELECT COUNT(*) as _textQty FROM texts WHERE
text_var='".$testtext.'";
    $res = mysql_query($query);
    if ($res){
        while($row = mysql_fetch_array($res)) {
            $result=$row['_textQty'];
        }
    }
    return $result;
}
```

Знаходимо індекс тексту \$testtext в БД:

```
function FindTextIndex($mysqli, $testtext) {
    $query="SELECT MAX(id) as _textID FROM texts WHERE
text_var='".$testtext.'";
    $res = mysql_query($query);
    if ($res){
        while($row = mysql_fetch_array($res)) {
            $result=$row['_textID'];
        }
    }
    return $result;
}
```

При заповненні форм для створення нової анкети формується JSON запит, цей запит скрипт переводить в асоціативний масив.

Отримати JSON як строку:

```
$json_str = file_get_contents('php://input');
```

Отримати об'єкт:

```
$json_obj = json_decode($json_str);
```

```
$anketeName = $json_obj->Name;
```

```
$QuestionsQty = $json_obj->QuestionsQty;
```

```
$NumberOfRespondents = $json_obj->NumberOfRespondents;
```

```
$Questions = $json_obj->Questions;//масив з питаннями
```

Далі створюємо анкету в БД з використанням описаних функцій та масиву отриманого з JSON запиту

```
//создаем новую анкету
```

```

//проверяем наличие анкеты с таким же именем
if (FindAnketeIndex($mysqli, $anketeName)>0) {
    echo("ERROR Ankete with this Name allrady exists");
    exit();
}
$MaxAnketeIndex=GetMaxAnketeIndex($mysqli);//максимальный индекс
анкеты в БД
$MaxAnketeIndex++;
AddNewAnkete($mysqli,$MaxAnketeIndex, $anketeName, 1,
$NumberOfRespondents);
//создаем новые вопросы
$MaxTextIndex=GetMaxTextIndex($mysqli);//максимальный индекс текста в
БД
$MaxQuestionIndex=GetMaxQuestionIndex($mysqli);//максимальный индекс
вопроса в БД
$MaxAnswerIndex=GetMaxAnswerIndex($mysqli);//максимальный индекс
ответа в БД
for ($i=0; $i<$QuestionsQnty; $i++) { //перебираем вопросы
    $QuestNumb=$Questions[$i]->Number;
    $QuestionText = $Questions[$i]->Text;
    $AnswersQnty = $Questions[$i]->AnswersQnty;
    $MaxAnswersTogether = $Questions[$i]->MaxAnswersTogether;
    $Answers = $Questions[$i]->Answers;
    if (TestTextAvailability($mysqli, $QuestionText)<1) { //если текст ранее
не встречался – добавляем
        $MaxTextIndex++;
        AddNewText($mysqli,$MaxTextIndex,
$QuestionText);//добавляем новый текст
        $MaxQuestionIndex++;
        AddNewQuestion($mysqli,$MaxQuestionIndex,
$MaxAnketeIndex, $MaxTextIndex, $MaxAnswersTogether);//добавляем новый вопрос
    }
    else { //если такой текст уже был – указываем его индекс
        $ThisTextIndex = FindTextIndex($mysqli,
$QuestionText);//Находим индекс текста в БД
        $MaxQuestionIndex++;
        AddNewQuestion($mysqli,$MaxQuestionIndex,
$MaxAnketeIndex, $ThisTextIndex, $MaxAnswersTogether);//добавляем новый вопрос
    }
}
//создаем новые ответы
for ($j=0; $j<$AnswersQnty; $j++) { //перебираем ответы
    $AnswNumb = $Answers[$j]->AnswNumber;
    $AnswText = $Answers[$j]->AnswText;
    $AnswType = $Answers[$j]->Type;
    if (TestTextAvailability($mysqli, $AnswText)<1) { //если текст
ранее не встречался – добавляем
        $MaxTextIndex++;
        AddNewText($mysqli,$MaxTextIndex,
$AnswText);//добавляем новый текст
        $MaxAnswerIndex++;
        AddNewAnswer($mysqli,$MaxAnswerIndex,
$MaxQuestionIndex, $AnswType, $MaxTextIndex);//добавляем новый ответ
    }
}

```

```

else { //если такой текст уже был – указываем его индекс
    $ThisTextIndex = FindTextIndex($mysqli,
$AnswText);//Находим индекс текста в БД
    $MaxAnswerIndex++;
    AddNewAnswer($mysqli,$MaxAnswerIndex,
$MaxQuestionIndex, $AnswType, $ThisTextIndex);//добавляем новый ответ
    }
}
}

```

Для проведення тестування було написано скрипт, що виводить зі створених нами додаткових таблиць бази даних тест посторінково TakeTest.php.

Загальний принцип роботи цього скрипта полягає у виведенні тексту запитання та елементів форми типу перемикач `<input type="radio" id="AnswerX">` та знизу виводиться кнопка як елемент форми `input type="submit"`, що дозволяє перейти на наступне питання або закінчити тестування, якщо це останнє питання:

```
echo('<input type="submit" value="'. $s_temp .'></p></form>');
```

Для отримання результатів тестування було також написано окремий скрипт, що дозволяє з таблиць з результатами тестування формувати звіти Results.php.

Загальний принцип роботи скрипта у тому, що для кожної заповненої анкети визначається учень за його ІД, що її заповнив та йому на сторінку результатів виводиться кількість правильних відповідей завдяки SQL запиту до таблиці результуючих відповідей.

3.5 Реалізація автоматичної перевірки написаного учнями програмного коду

Одним з елементів інтерактивності розробленого онлайн курсу є можливість автоматичної перевірки правильності написання учнем коду програми. Тобто ми реалізуємо синтаксичний аналізатор мови програмування C++ з використанням PHP скрипта.

Основа роботи синтаксичного аналізатору полягає в збереженні тексту, що має пройти перевірку у текстовому файлі та запуск скрипта з построкковим аналізом тексту у цьому файлі.

Найбільш поширені помилки новачка при написанні програмного коду:

- відсутні крапки з комою; для завершення операторів / рядків;
- невідповідні рядкові лапки;
- забуті оператори, зокрема для рядка конкатенації;
- незбалансовані дужки;

При аналізі треба виходити з наступних правил та їх послідовності:

а) видаляємо та запам'ятовуємо де били розташовані всі коментарі, щоб не приймати їх в аналіз. тобто відшукуємо символи «//» та видаляємо з рядка все що після них та відшукуємо пару «/**» та «**/» та видаляємо все що між ними;

б) відшукуємо всі відкриті дужки як круглі так і фігурні та квадратні. якщо кількість відкритих дужок не відповідає кількості закритих треба запам'ятати рядок та позицію дужок без пари для подальшого підсвітлення їх червоним кольором при виведенні тексту програми після перевірки;

в) виявлення правильності вказання послідовності аргументів в операторах. оскільки ми працюємо лише з вузькою кількістю команд в курсі наведемо їх перелік та правила (табл.. 3.2);

г) перевірка наявності крапки з комою в кінці кожного рядка та підсвічування всього рядка де таке виявлено;

г) для оператору перевірки if перевіряємо правильність розташування умови або взаємодій між умовами коли їх декілька, а також наявність фігурних дужок після оператору;

д) в умовах перевірки перевіряємо правильність операторів «==», «>=», «<=», «!=», «>», «<» та умов взаємодії «||», «&&»;

е) перевіряємо наявність декількох підряд однакових знаків пунктуації;

ж) перевіряємо наявність позначення типів для всіх змінних, що використовуються в програмі;

з) перевіряємо правильність послідовності тексту. спочатку автозаміни та змінні, потім процедура налаштування, потім основний цикл.

Таблиця 3.2 – Правила для основних операторів мови C++, що перевіряються скриптом

Оператор	Правило	Приклад
Присвоєння «=»	[тип змінної] [ім'я змінної] = [значення];	int c = 5; b = 2.8 (якщо тип b описано вище)
Зчитати значення з ніжки плати ардуіно «digitalRead»	digitalRead ([номер або ім'я ніжки]);	digitalRead(2); digitalRead(Input1); якщо Input1 визначено раніше
Записати значення на ніжку плати ардуіно «digitalWrite»	digitalWrite ([номер або ім'я ніжки]);	digitalWrite(5); digitalWrite(pin5); якщо pin5 визначено раніше
Пауза в мілісекундах «delay»	delay ([тривалість затримки в мс]);	delay(200); delay(pause);
Пауза в мікросекундах «delayMicroseconds»	delayMicroseconds ([тривалість затримки в мкс]);	delayMicroseconds(400); delayMicroseconds(pause0);
Перевірка «if»	if ([умова1]) [взаємодія умов]([умова2]) і так далі (якщо більше 2х) {[інші оператори]}	if (b==5) {...} if ((c>=)&&(b<5) {...}

Продовження таблиці 3.2

Об'явлення типу змінної	[тип змінної][імя змінної];	byte b; int c;
Автозаміна символу в тексті «#define»	#define [ім'я на яке заміняємо] [що заміняємо]	#define input1 3
Процедура налаштування «setup»	void setup() { [послідовність операторів] }	void setup() {....}
Основний цикл програми «loop»	void loop() { [послідовність операторів] }	void loop() {.....}

3.6 Висновки до третього розділу

У третьому розділі проводилась розробка програмного коду системи. На першому етапі було розроблено інтерфейс навчального курсу на основі системи керування вмістом, що включає в себе розробку власної теми оформлення, власних меню та блоків.

Далі у розділі було проведено розробку навчального контексту та опис його розміщення на сайті. З використанням інструменту phpMyAdmin було створено додаткові таблиці бази даних та описано їх застосування. Також в даному розділі проводився опис процесу написання програмного коду навчального курсу.

ВИСНОВКИ

У магістерській роботі було розроблено інтерактивний курс з інформатики для учнів 7 класу.

Навчальний курс представляє собою інтернет сайт з широким колом можливостей, інтерфейс сайту написаний на CMS Drupal, а додатковий функціонал сайту реалізований за допомогою скриптів, що написані на PHP.

Навчальний курс містить:

- навчальні матеріали з загального курсу інформатики згідно підручників затверджених Міністерством освіти та науки України;
- навчальні матеріали для отримання початкових даних з програмування платформи Arduino на мові програмування C++ з використанням початкових навичок в робототехніці;
- можливість ствернення тестів для автоматичної перевірки отриманих на протязі вивчення курсу знань учнів з бальним оцінюванням;
- можливість перевірки правильності написання програмного коду безпосередньо на сайті під час вивчення відповідних тем;
- можливість швидкої взаємодії учнів з вчителем за рахунок реалізації відправлення повідомлень на найпоширеніші меседжери.

Практична значимість роботи полягає в наступному:

- проведено порівняльний аналіз і отримана експертна оцінка поширених програмних засобів створення інтерактивних навчальних систем;
- спроектована, реалізована на базі PHP і впроваджена в навчальний процес інтерактивна система з дисципліни «Інформатика» для учнів 7-го класу.

ПЕРЕЛІК ПОСИЛАНЬ

1. Доррер А. Г., Иванилова Т. Н. Моделирование интерактивного адаптивного обучающего курса // *Современные проблемы науки и образования*. 2007. № 5. 96 с.
2. Двудличанская Н. Н. Интерактивные методы обучения как средство формирования ключевых компетенций // *Наука и образование: электронное научно–техническое издание*. 2011. 78 с.
3. Полат Е. С., Бухаркина М. Ю., Моисеева М. В., Петров А.Е. Новые педагогические и информационные технологии в системе образования: учебное пособие. Москва : Академия, 2000. 234 с.
4. Горбунова Л. И., Субботина Е. А. Использование информационных технологий в процессе обучения // *Молодой ученый*. 2013. №4. 547 с.
5. Лосик С. Н. Поиск путей реализации непрерывного курса информатики: подходы к обучению учащихся в начальной школе // *Интерактивное образование*. 2018. №1. 243 с.
6. Босова Л. Л. Метапредметная направленность – одна из основных характеристик пропедевтического этапа школьного курса информатики // *Вестник Северо–восточного федерального университета им. М. К. Аммосова*. 2009. No 4. Т. 6. 53 с.
7. Каплан А. В. Результаты апробации учебно–методического комплекта «Информатика для всех» в первом классе // *Информатика в школе*. 2017. № 3. 55 с.
8. Павлов Д. И. Раскрытие содержательных линий «Представление информации» и «Информационных процессов» на уровне начального общего образования // *Открытое и дистанционное образование*. Томск, 2018. № 1. 66 с.

9. Гальцова З. Д. Информатика доступна всем // *Педагогика и психология // Перспективы развития: материалы II Междунар. науч.–практ. конференции*. Чебоксары : ЦНС «Интерактив плюс», 2017. 88 с.

10. Деркач Т. Л., Саприкіна М. С. Твоє майбутнє у сфері інформаційних технологій. Київ : Видавництво «Юстон», 2018. 40 с.

11. Ракитина Е. А. Построение методической системы обучения информатике на деятельностной основе : Дис. д-ра пед. наук. Москва, 2002. 485 с.

12. Освітній Портал TeachPro URL: <http://www.teachpro.ru/> (дата звернення: 25.11.2019).

13. Центр дистанційної освіти Ейдос URL: <https://eidos.ru/> (дата звернення: 25.11.2019).

14. Онлайн – школа «Фоксфорд» URL: <https://foxford.ru/> (дата звернення: 25.11.2019).

15. Prometheus – український проект масових відкритих онлайн–курсів URL: <https://prometheus.org.ua/> (дата звернення: 26.11.2019).

16. Дистанційні курси «ДистОсвіта» URL: <https://dystosvita.gnomio.com/> (дата звернення: 26.11.2019).

17. Ривкінд Й. Я. Информатика: підручник для 7–го класу загальноосвітніх навчальних закладів. Київ : Генеза, 2015. 40с.

18. Гуржій А. М., Карташова Л. А., Лапінський В. В., Руденко В. Д. Информатика: підручник для 7–го класу загальноосвітніх навчальних закладів. Львів : Світ, 2015. 176 с.

ДОДАТОК А

Код файлу NewAnkete.php

```

<?php
    header("Content-Type: text/html; charset=windows-1251");
    include 'Settings.php';

    $MaxAnketeIndex=0;//максимальный индекс анкеты в БД
    $MaxTextIndex=0;//максимальный индекс текста в БД
    $MaxQuestionIndex=0;//максимальный индекс вопроса в БД
    $MaxAnswerIndex=0;//максимальный индекс ответат в БД

//преобразовываем полученный запрос в массив
// Получить JSON как строку
    $json_str = file_get_contents('php://input');
//Получить объект
    $json_obj = json_decode($json_str);
    $anketeName = $json_obj->Name;
    $QuestionsQty = $json_obj->QuestionsQty;
    $NumberOfRespondents = $json_obj->NumberOfRespondents;
    $Questions = $json_obj->Questions;//массив с вопросами
    saveLog(" ".$anketeName." ".$QuestionsQty." ".$NumberOfRespondents);!!!!!!

//соединяемся с БД
    $dbcnx = @mysql_connect($dblocation,$dbuser,$dbpasswd);
    if (!$dbcnx) // Если дескриптор равен 0 соединение не установлено
    {
        echo("ERROR_DB_CONNECT_0<br>");
        exit();
    }
    if ($dbcnx>0)
    {
        //echo("DB_CONNECT_OK<br>");
    }
// Код соединения с базой данных
    if (@mysql_select_db($dbname, $dbcnx))
    {
        echo( "ERROR_DB_CONNECT_1<br>" );
        exit();
    }
//Указываем кодировку для записи в БД
    mysql_query("SET NAMES UTF8");
    mysql_query("SET CHARACTER SET UTF8");
//создаем новую анкету
    //проверяем наличие анкеты с таким же именем
    if (FindAnketeIndex($mysqli, $anketeName)>0) {
        echo("ERROR Ankete with this Name allrady exists");
        exit();
    }
    $MaxAnketeIndex=GetMaxAnketeIndex($mysqli);//максимальный индекс анкеты в БД
    $MaxAnketeIndex++;
    AddNewAnkete($mysqli,$MaxAnketeIndex, $anketeName, 1, $NumberOfRespondents);
//создаем новые вопросы
    $MaxTextIndex=GetMaxTextIndex($mysqli);//максимальный индекс текста в БД
    $MaxQuestionIndex=GetMaxQuestionIndex($mysqli);//максимальный индекс вопроса в БД
    $MaxAnswerIndex=GetMaxAnswerIndex($mysqli);//максимальный индекс ответат в БД
    for ($i=0; $i<$QuestionsQty; $i++) { //перебираем вопросы

```

```

$QuestNumb=$Questions[$i]->Number;
$QuestionText = $Questions[$i]->Text;
$AnswersQnty = $Questions[$i]->AnswersQnty;
$MaxAnswersTogether = $Questions[$i]->MaxAnswersTogether;
$Answers = $Questions[$i]->Answers;
if (TestTextAvailability($mysqli, $QuestionText)<1) { //если текст ранее не встречался –
добавляем
    $MaxTextIndex++;
    AddNewText($mysqli,$MaxTextIndex, $QuestionText); //добавляем новый текст
    $MaxQuestionIndex++;
    AddNewQuestion($mysqli,$MaxQuestionIndex, $MaxAnketeIndex, $MaxTextIndex,
$MaxAnswersTogether); //добавляем новый вопрос
}
else { //если такой текст уже был – указываем его индекс
    $ThisTextIndex = FindTextIndex($mysqli, $QuestionText); //Находим индекс текста в
БД
    $MaxQuestionIndex++;
    AddNewQuestion($mysqli,$MaxQuestionIndex, $MaxAnketeIndex, $ThisTextIndex,
$MaxAnswersTogether); //добавляем новый вопрос
}
//создаем новые ответы
for ($j=0; $j<$AnswersQnty; $j++) { //перебираем ответы
    $AnswNumb = $Answers[$j]->AnswNumber;
    $AnswText = $Answers[$j]->AnswText;
    $AnswType = $Answers[$j]->Type;
    if (TestTextAvailability($mysqli, $AnswText)<1) { //если текст ранее не встречался –
добавляем
        $MaxTextIndex++;
        AddNewText($mysqli,$MaxTextIndex, $AnswText); //добавляем новый текст
        $MaxAnswerIndex++;
        AddNewAnswer($mysqli,$MaxAnswerIndex, $MaxQuestionIndex,
$AnswType, $MaxTextIndex); //добавляем новый ответ
    }
    else { //если такой текст уже был – указываем его индекс
        $ThisTextIndex = FindTextIndex($mysqli, $AnswText); //Находим индекс
текста в БД
        $MaxAnswerIndex++;
        AddNewAnswer($mysqli,$MaxAnswerIndex, $MaxQuestionIndex,
$AnswType, $ThisTextIndex); //добавляем новый ответ
    }
}
}
echo("Ankete SENDED OK");

//узнаем максимальный индекс анкет на сервере
function GetMaxAnketeIndex($mysqli){
    $query="SELECT MAX(id) as _maxID FROM questionnaires;";
    $res = mysql_query($query);
    if ($res){
        while($row = mysql_fetch_array($res)) {
            $result=$row['_maxID'];
        }
    }
    return $result;
}

//узнаем максимальный индекс текста на сервере
function GetMaxTextIndex($mysqli){
    $query="SELECT MAX(id) as _maxID FROM texts;";
    $res = mysql_query($query);
    if ($res){
        while($row = mysql_fetch_array($res)) {
            $result=$row['_maxID'];
        }
    }
}

```

```

        return $result;
    }
    //узнаем максимальный индекс вопроса на сервере
    function GetMaxQuestionIndex($mysqli){
        $query="SELECT MAX(id) as _maxID FROM questions;";
        $res = mysql_query($query);
        if ($res){
            while($row = mysql_fetch_array($res)) {
                $result=$row['_maxID'];
            }
        }
        return $result;
    }
    //узнаем максимальный индекс ответа на сервере
    function GetMaxAnswerIndex($mysqli){
        $query="SELECT MAX(id) as _maxID FROM answers;";
        $res = mysql_query($query);
        if ($res){
            while($row = mysql_fetch_array($res)) {
                $result=$row['_maxID'];
            }
        }
        return $result;
    }
    //добавляем новый текст с индексом в БД
    function AddNewText($mysqli,$textid, $newtext){
        $query="INSERT INTO texts VALUES (".$textid.", " ."$newtext."");";
        $res = mysql_query($query);
    }
    //добавляем новую анкету с индексом в БД
    function AddNewAnkete($mysqli,$anketeid, $name, $author, $respondents){
        $datenow=date('Y-m-d', time());
        $query="INSERT INTO questionnaires VALUES (".$anketeid.", " ."$name.", " ."$author.", " ."$datenow.", " ."$respondents."");";
        $res = mysql_query($query);
    }
    //добавляем новый вопрос с индексом в БД
    function AddNewQuestion($mysqli,$quest_id, $ankete_id, $qtext_id, $answers_lim){
        $query="INSERT INTO questions VALUES(".$quest_id.", ".$ankete_id.", 0, 0, ".$qtext_id.", " ."$answers_lim."");";
        $res = mysql_query($query);
    }
    //добавляем новый ответ с индексом в БД
    function AddNewAnswer($mysqli,$answ_id, $quest_id, $answ_type, $qtext_id){
        $query="INSERT INTO answers VALUES(".$answ_id.", ".$quest_id.", " ."$answ_type.", " ."$qtext_id."");";
        $res = mysql_query($query);
    }
    //Проверяем наличие текста в БД – возвращает сколько раз текст встречается в БД
    function TestTextAvailability($mysqli, $stesttext){
        $query="SELECT COUNT(*) as _textQty FROM texts WHERE text_var="."$stesttext."";";
        $res = mysql_query($query);
        if ($res){
            while($row = mysql_fetch_array($res)) {
                $result=$row['_textQty'];
            }
        }
        return $result;
    }
    //Находим индекс текста в БД
    function FindTextIndex($mysqli, $stesttext) {
        $query="SELECT MAX(id) as _textID FROM texts WHERE text_var="."$stesttext."";";
        $res = mysql_query($query);
        if ($res){
            while($row = mysql_fetch_array($res)) {

```

```
                $result=$row['_textID'];
            }
        }
    }
    return $result;
}
//Находим индекс анкеты с указанным именем в БД
function FindAnketeIndex($mysqli, $testname) {
    $query="SELECT MAX(id) as _anketeID FROM questionnaires WHERE Name=".". $testname. ".";
    $res = mysql_query($query);
    if ($res){
        while($row = mysql_fetch_array($res)) {
            $result=$row['_anketeID'];
        }
    }
    return $result;
}
?>
```

ДОДАТОК Б

Код файла Results.php

```

<?php
    header("Content-Type: text/html; charset=windows-1251");
    include 'Settings.php';
    $NeedSaveToFile = 0;//необходимость сохранения результатов в файл 0 – нет, 1 – да
    $NeedAnketeText = 0;//необходимость вывода содержимого анкеты на страницу
    $NeedAnketesList = 1;//необходимость вывода перечня опрошенных анкет на страницу
    $NeedResults = 0;//необходимость вывода результатов в виде таблиц на страницу
    $NeedSampling = 0;//необходимость вывода выборки

//соединение с БД
    $dbcnx = @mysql_connect($dblocation,$dbuser,$dbpasswd);
    if (!$dbcnx) // Если дескриптор равен 0 соединение не установлено
    {
        echo("ERROR_DB_CONNECT_0<br>");
        exit();
    }
    if ($dbcnx>0)
    {
        //echo("DB_CONNECT_OK<br>");
    }
// Код соединения с базой данных
    if (!@mysql_select_db($dbname, $dbcnx))
    {
        echo( "ERROR_DB_CONNECT_1<br>" );
        exit();
    }
//выводим список анкет с названиями
$res = mysql_query("SELECT id, Name FROM questionnaires;");
    if ($res){
        $i=0;
        while($row = mysql_fetch_array($res)) {
            $i++;
            echo ($row['id']. " | " . $row['Name']. "<br>");
        }
        $NumberOfQuestionnaires=$i;
        $s_temp="Всего опросов:";
        $s_temp=iconv("utf-8", "windows-1251", $s_temp);
        echo ($s_temp.$NumberOfQuestionnaires."<br>");
    }
    else {
        echo "ERROR_RESULTS_NOT_FOUND<br>";
        exit();
    }
//форма для ввода номера анкеты
    $s_temp="Выберите анкету:";
    $s_temp=iconv("utf-8", "windows-1251", $s_temp);
    $s_temp1="Формировать файл с результатами?";
    $s_temp1=iconv("utf-8", "windows-1251", $s_temp1);
    $s_temp2="Выводить вопросы анкеты?";
    $s_temp2=iconv("utf-8", "windows-1251", $s_temp2);
    $s_temp3="Выводить опрошенные анкеты?";
    $s_temp3=iconv("utf-8", "windows-1251", $s_temp3);
    $s_temp4="Выводить результаты в виде таблиц?";
    $s_temp4=iconv("utf-8", "windows-1251", $s_temp4);
?>

```

```

<form action="" method="post">
  <div>
    <input type="checkbox" name="needfile">
    <label for="needfile"><?php echo($s_temp1) ?></label>
  </div>
  <div>
    <input type="checkbox" name="needtext">
    <label for="needtext"><?php echo($s_temp2) ?></label>
  </div>
  <div>
    <input type="checkbox" name="needlist">
    <label for="needlist"><?php echo($s_temp3) ?></label>
  </div>
  <div>
    <input type="checkbox" name="needresult">
    <label for="needresult"><?php echo($s_temp4) ?></label>
  </div>
  <p><?php echo($s_temp) ?> <input type="text" name="AnketeID" />
  <input type="submit" value="OK"/></p>
</form>

<?php
//Проверяем заполнена ли форма – если заполнена выводим ее результаты
if (isset($_POST['AnketeID'])&&((int)$_POST['AnketeID']>0)) {
  //$NeedSaveToFile = isset($_POST['needfile'])? 1 : 0;
  $NeedSaveToFile = 1;
  $NeedAnketeText = isset($_POST['needtext'])? 1 : 0;
  $NeedAnketesList = isset($_POST['needlist'])? 1 : 0;
  $NeedResults = isset($_POST['needresult'])? 1 : 0;

  $s_temp="Выбранная анкета:";
  $s_temp=iconv("utf-8", "windows-1251", $s_temp);
  echo ($s_temp.(int)$_POST['AnketeID']."<br>");
  $idAnkete=(int)$_POST['AnketeID'];
  if ($idAnkete>$NumberOfQuestionnaires) {
    $s_temp="Анкеты с таким номером не существует! Выберите другой номер анкеты";
    $s_temp=iconv("utf-8", "windows-1251", $s_temp);
    echo ($s_temp);
    exit();}
  //ТУТ обработка и вывод результатов по выбранной анкете
  //определяем общее количество заполненных анкет в данном опросе
  $res = mysql_query("SELECT Count(*) as _TotalAnswers FROM res_questionnaires WHERE
idQuestionnaires=".$idAnkete."");
  if ($res){
    while($row = mysql_fetch_array($res)) {
      $FinishedAnketes=$row['_TotalAnswers'];
    }
    $s_temp="Всего опрошено интервьюеров: ";
    $s_temp=iconv("utf-8", "windows-1251", $s_temp);
    echo ($s_temp.$FinishedAnketes."<br>");
  }
  else {
    echo "NO INTERVIEWERS FOUNDED<br>";
    exit();
  }
  //определяем список вопросов и выводим шапку таблицы с их номерами внутри анкеты
  $s_temp="Список вопросов анкеты: ";
  $s_temp=iconv("utf-8", "windows-1251", $s_temp);
  if ($NeedAnketeText == 1) {echo ($s_temp."<br>");}
  //достаем список вопросов в 2 динамических массива с ИД вопроса и текста – номер элемента
  соответствует номеру вопроса в анкете
  $QuestionID = array();//массив ИД вопросов
  $QuestionText = array();//массив текстовок вопросов
  $QuestAnswCnt = array();//массив кол-ва разрешенных одновременных ответов

```



```

$res = mysql_query("SELECT questions.id as _QuestID, texts.text_var as _QuestText,
questions.Answ_count as _AnswCnt FROM questions, texts WHERE questions.idQuestionnaire=".SidAnkete."
AND texts.id=questions.idText;");
    if ($res){
        while($row = mysql_fetch_array($res)) {
            $QuestionID[]=$row['_QuestID'];
            $QuestionText[]=$row['_QuestText'];
            $QuestAnswCnt[]=$row['_AnswCnt'];
        }
    }
    else {
        echo "NO QUESTIONS FOUNDED <br>";
        exit();
    }
    //достаем список ответов в 2 димамических массива с ИД вопроса! и текста
    //и добавляем 3й массив с номером ответа в вопросе
    $AnswQuestID = array();//ИД вопроса к которому привязан ответ
    $AnswerText = array();//текст ответа
    $AnswNumbInQuest = array(); //номер ответа в вопросе
    $AnswID = array(); //ИД ответа
    for ($i=0 ; $i<count($QuestionID); $i++) {
        $AnswQuery = "SELECT DISTINCT answers.idQuestion as _AnswQuestID, answers.id as
_AnswID, texts.text_var as _AnswText FROM answers, questions, texts WHERE
questions.idQuestionnaire=".SidAnkete." AND texts.id=answers.idText AND
answers.idQuestion=".$QuestionID[$i].".";";
        $res = mysql_query($AnswQuery);
        if ($res){
            $j=0;
            while($row = mysql_fetch_array($res)) {
                $j++;
                $AnswQuestID[]=$row['_AnswQuestID'];
                $AnswerText[]=$row['_AnswText'];
                $AnswID[]=$row['_AnswID'];
                $AnswNumbInQuest[]=$j;
            }
        }
        else {
            echo "NO ANSWERS FOUNDED <br>";
            exit();
        }
    }
    //выводим на страницу вопросы и ответы т.е. полную анкету
    if ($NeedAnketeText == 1) {
        for ($i=0 ; $i<count($QuestionID); $i++){
            //echo(($i+1)." ID=".$QuestionID[$i]." ".$QuestionText[$i]."<br>");
            echo(($i+1)." ".$QuestionText[$i]."<br>");
            for ($j=0 ; $j<count($AnswID); $j++){
                //if ($AnswQuestID[$j]==$QuestionID[$i])
                {echo("&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;".$AnswNumbInQuest[$j]." ID=".$AnswID[$j]." Question
ID=".$AnswQuestID[$j]." ".$AnswerText[$j]."<br>");}
                if ($AnswQuestID[$j]==$QuestionID[$i])
                {echo("&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;".$AnswNumbInQuest[$j].") ".$AnswerText[$j]."<br>");}
            }
        }
    }
    //Заполняем массив собственных вариантов ответа для текущей анкеты
    $OwnAnswersID = array();//массив ИД собственных вариантов ответа
    $OwnAnswersText = array();//массив текстов собственных вариантов ответа
    $OwnAnswQuestID = array();//массив ИД вопросов к которому относится собственный ответ
    $QueryWhat = "res_texts.id as _OwnAnswID, res_texts.text_var as _OwnAnswText, answers.id as
_OwnAnswQustID ";
    $QueryFrom = " FROM res_questionnaires, res_questions, res_answers, answers, res_texts";
    $QueryWhere = " WHERE res_questionnaires.idQuestionnaires=".SidAnkete;

```

```

$QueryWhere = $QueryWhere." AND res_questions.idRes_Questionnaire=res_questionnaires.id AND
res_questions.id=res_answers.idRes_Question";
$QueryWhere = $QueryWhere." AND answers.id=res_answers.idAnswer AND answers.answer_type=1";
$QueryWhere = $QueryWhere." AND res_texts.id=res_answers.idRes_text AND res_answers.idRes_text
IS NOT NULL";
$QueryOrder = " ORDER BY answers.id;";
$TakeAnswers="SELECT DISTINCT ".$QueryWhat.$QueryFrom.$QueryWhere.$QueryOrder;
$res = mysql_query($TakeAnswers);
    if ($res){
        //j = 0;////
        while($row = mysql_fetch_array($res)) {
            $OwnAnswersID[]=$row['_OwnAnswID'];
            $OwnAnswersText[]=$row['_OwnAnswText'];
            $OwnAnswQuestID[]=$row['_OwnAnswQustID'];
            //echo (($j+1)." ".$OwnAnswersID[$j]." ".$OwnAnswersText[$j].<br>);////
            //j++;////
        }
    }
    else {
        echo "NO OWN ANSWERS FOUNDED <br>";
        exit();
    }
}
//Выводим шапку таблицы заполненных анкет
if ($NeedAnketesList == 1) {
    $s_temp="Опрошенные анкеты:";
    $s_temp=iconv("utf-8", "windows-1251", $s_temp);
    $s_temp2="Кто";
    $s_temp2=iconv("utf-8", "windows-1251", $s_temp2);
    $s_temp3="Время";
    $s_temp3=iconv("utf-8", "windows-1251", $s_temp3);
    $s_temp4="№";
    $s_temp4=iconv("utf-8", "windows-1251", $s_temp4);
    echo (<table border="1" width="100%">);
    echo (<caption>".$s_temp.</caption>);
    echo (<thead>);
    echo (<th width="3%">'.$s_temp4.</th>);
    echo (<th width="3%">'.$s_temp2.</th>);
    echo (<th width="6%">'.$s_temp3.</th>);
    $procent = round(100/count($QuestionID)*0.88);//0.88 = 100%-3%-3%-6%
    //echo (count($QuestionID)." ".$procent);
    for ($i=0 ; $i<count($QuestionID); $i++){
        echo (<th width="".$procent.%">'.$(i+1).</th>);
    }
    echo (</thead> );
}
//Выводим результаты заполненных анкет
$QueryWhat = "res_questions.idQuestion as _questionID, texts.text_var as _answer,
res_questionnaires.idUser as _user";
$QueryWhat = $QueryWhat.", res_questionnaires.Create_Date as _time, res_questionnaires.id as
_CompletedAnketeID";
$QueryWhat = $QueryWhat.", res_answers.idRes_text as _UniqAnswer, res_answers.idAnswer as
_answerID";
$QueryFrom = " FROM res_questionnaires, res_questions, res_answers, answers, texts";
$QueryWhere = " WHERE res_questionnaires.idQuestionnaires=".$SidAnkete;
$QueryWhere = $QueryWhere." AND res_questions.idRes_Questionnaire=res_questionnaires.id AND
res_questions.id=res_answers.idRes_Question";
$QueryWhere = $QueryWhere." AND answers.id=res_answers.idAnswer AND answers.idText=texts.id ";
$QueryOrder = " ORDER BY res_questionnaires.id;";
$TakeAnswers="SELECT DISTINCT ".$QueryWhat.$QueryFrom.$QueryWhere.$QueryOrder;
//параллельно пишем в файл
$name="Results/Res".date('mdY_his', time()).".txt"; //название файла с датой и временем среза
$str="*Total Completed Anketes:".$FinishedAnketes; //количество заполненных анкет
if ($NeedSaveToFile == 1) { WriteStrToFile($str, $name); }
//расшифровка кодов вопросов

```

```

$str="*Questions:".count($QuestionID); //количество вопросов в анкете
if ($NeedSaveToFile == 1) {
    WriteStrToFile($str, $fname);
    for ($k=0; $k<(count($QuestionID)); $k++) { //записываем расшифровку
        $str=$QuestionID[$k]."|".$QuestAnswCnt[$k]."|".$QuestionText[$k];
        WriteStrToFile($str, $fname);
    }
}
//расшифровка кодов ответов
$str="*Answers:".count($AnswID); //общее количество ответов в анкете
if ($NeedSaveToFile == 1) {
    WriteStrToFile($str, $fname);
    for ($k=0; $k<(count($AnswID)); $k++) { //записываем расшифровку
        $str=$AnswID[$k]."|".$AnswQuestID[$k]."|".$AnswerText[$k];
        WriteStrToFile($str, $fname);
    }
}
//расшифровка кодов собственных ответов
$str="*Own Answers:".count($OwnAnswersID); //общее количество ответов в анкете
if ($NeedSaveToFile == 1) {
    WriteStrToFile($str, $fname);
    //echo ("*Own Answers:".count($OwnAnswersID)."<br>");
    for ($k=0; $k<(count($OwnAnswersID)); $k++) { //записываем расшифровку
        $str=$OwnAnswersID[$k]."|".$OwnAnswQuestID[$k]."|".$OwnAnswersText[$k];
        WriteStrToFile($str, $fname);
    }
}
if ($NeedSaveToFile == 1) { WriteStrToFile("*Results", $fname); }
//результаты анкетирования текстом
//echo($TakeAnswers."<br>");//!!!
$res = mysql_query($TakeAnswers);
if ($res){
    //echo("Reading results... <br>");//!!!!
    $i=1; //номер заполненной анкеты
    $AnketeStr = array(); //массив содержащий текущую заполненную анкету для вывода в
таблицу
    $AnketeStrID = array(); //массив содержащий текущую заполненную анкету в виде ИД для
записи в файл
    for ($k=0; $k<(count($QuestionID)+3); $k++) { //заполняем массив пробелами
        $AnketeStr[$k] = " ";
    }
    for ($k=0; $k<(count($QuestionID)); $k++) { //заполняем массив 0
        $AnketeStrID[$k] = 0;
    }
    $ThisAnketeUser = 0;
    $ThisAnketeTime = 0;
    $PrevQuest = 0; //ИД вопроса для предыдущего ответа
    $ThisQuest = 0; //ИД вопроса текущего ответа
    //нужно чтобы выводить в одной ячейке в случае когда несколько вариантов ответов
выбрано для вопроса
    $PrevAnkete = 0; //ИД предыдущей заполненной анкеты
    $ThisAnkete = 0; //ИД текущей заполненной анкеты
    //нужно чтоб понимать когда переходим к следующей строке таблицы
    while($row = mysql_fetch_array($res)) {
        $UniqueAnswer = $row['_UniqAnswer']; //текст для своего варианта ответа !!!пока
это индекс текста
        $ThisAnkete = $row['_CompletedAnketeID']; //ИД заполненной анкеты
        $ThisQuest = $row['_questionID']; //ИД заполненного ответа
        //если это новая заполненная анкета и предыдущая > 0 – выводим предыдущую
строку и обновляем массив строки
        if (($ThisAnkete != $PrevAnkete) && ($PrevAnkete > 0)){
            if ($NeedAnketesList == 1) {
                echo '<tr>';
                echo '<td align="right">'. $i. '</td>'; //номер заполненной анкеты

```

```

    echo '<td align="right">'. $ThisAnketeUser. '</td>'; //ИД пользователя
который заполнил
    echo '<td align="center">'. $ThisAnketeTime. '</td>'; //время окончания
заполнения
}
$str = "";
for ($k=0; $k<(count($QuestionID)); $k++) {
    if ($NeedAnketesList == 1) { echo ('<td
align="left">'. $AnketeStr[$k+3]. '</td>');
        $str=$str.$AnketeStrID[$k]. "|";
    }
    if ($NeedAnketesList == 1) {
        echo '</tr>';
    }
    if ($NeedSaveToFile == 1) { WriteStrToFile($str, $fname); }
    for ($k=0; $k<(count($QuestionID)+3); $k++) { //заполняем массив пробелами
        $AnketeStr[$k] = " ";
    }
    for ($k=0; $k<(count($QuestionID)); $k++) { //заполняем массив 0
        $AnketeStrID[$k] = 0;
    }
    $i++;
}
$ThisAnketeUser = $row['_user'];
$ThisAnketeTime = $row['_time'];
$AnketeStr[0] = $i;
$AnketeStr[1] = $ThisAnketeUser;
$AnketeStr[2] = $ThisAnketeTime;
$PrevAnkete = $ThisAnkete;
//ищем какому номеру вопроса соответствует текущий ответ
$n = -1;
for ($k=0; $k<(count($QuestionID)); $k++) {
    if ($ThisQuest != $QuestionID[$k]) { continue; }
    else $n = $k;
}
if ($n>-1) {
    $ThisAnswer = $row['_answer'];
    $ThisAnswerID = $row['_answerID'];
    if ($UniqueAnswer>"") {
        // $ThisAnswerID = $ThisAnswerID. "_" . $UniqueAnswer;
        //ищем текст собственного варианта ответа из массива
        for ($k=0; $k<(count($OwnAnswersID)); $k++) {
            if ($UniqueAnswer == $OwnAnswersID[$k]) {
                $ThisAnswer = $OwnAnswersText[$k];
                $ThisAnswerID = $ThisAnswer;
            }
        }
    }
    if ($AnketeStr[$n+3] != " ") {
        $AnketeStr[$n+3] = $AnketeStr[$n+3]. " ". $ThisAnswer;
        $AnketeStrID[$n] = $AnketeStrID[$n]. " ". $ThisAnswerID;
    }
    else {
        $AnketeStr[$n+3] = $ThisAnswer;
        $AnketeStrID[$n] = $ThisAnswerID;
    }
}
}
// $PrevQuest = $ThisQuest;
}
//заполняем последнюю строчку
if ($NeedAnketesList == 1) {
    echo '<tr>';
    echo '<td align="right">'. $i. '</td>';
    echo '<td align="right">'. $ThisAnketeUser. '</td>';
}

```

```

echo '<td align="center">'. $ThisAnketeTime. '</td>';
$str = "";
for ($k=0; $k<(count($QuestionID)); $k++) {
    echo ('<td align="left">'. $AnketeStr[$k+3]. '</td>');
    $str=$str.$AnketeStrID[$k]. "|";
}
}
if ($NeedSaveToFile == 1) {WriteStrToFile($str, $fname);}
if ($NeedAnketesList == 1) {
    echo '</tr>';
    echo '</table>';
}
}
else {
    echo "ERROR_ANKETES_NOT_FOUND<br>";
    exit();
}
//Создаем одномерные таблицы с результатами по каждому вопросу
//Количество анкет, которому соответствует кол-во таблиц с одномерками
if ($NeedResults == 1) {
    for ($k=0; $k<(count($QuestionID)); $k++) {
        //делаем шапку таблицы
        $s_temp="<br>Вопрос №";
        $s_temp=iconv("utf-8", "windows-1251", $s_temp);
        $s_temp=$s_temp.($k+1). " ". $QuestionText[$k];
        echo ('<table border="1" width="100%">');
        echo ("<caption>". $s_temp. "</caption>");
        echo ("<thead>");
        //находим количество ответов в вопросе !!!пока без собственных
        $AnswersInQuestion = 0; //количество ответов для этого вопроса
        $FirstAnswerInQuest = -1; //индекс первого ответа для данного вопроса (должны
идти по порядку т.к. анкеты одновременно не создаются)
        for ($i=0 ; $i<count($AnswQuestID); $i++){//перебираем массив ответов у которых
ИД вопроса соответствует текущему вопросу
            if ($AnswQuestID[$i]==$QuestionID[$k]) {//если в ответе ИД вопроса
совпадает с ИД текущего вопроса
                $AnswersInQuestion++;//ответов в вопросе
                if ($FirstAnswerInQuest == -1) {$FirstAnswerInQuest = $i;}
//определяем индекс первого ответа для выяснения его кода
            }
        }
        $procent = round(100/$AnswersInQuestion);
        for ($i=$FirstAnswerInQuest; $i<($FirstAnswerInQuest+$AnswersInQuestion); $i++){
            echo ('<th width="'. $procent. "%">'. $AnswerText[$i]. '</th>');
        }
        echo ('</thead>');
        //заполняем количество ответов
        echo '<tr>'; //начало строки с количеством ответов
        $QuestSumma = 0; //общая сумма ответов
        $QuestResults = array();//создаем массив пустой результатов
        for ($i=$FirstAnswerInQuest; $i<($FirstAnswerInQuest+$AnswersInQuestion); $i++){
            $ThisAnswVal = $AnswID[$i];
            //делаем запрос к БД и определим кол-во ответов с ИД $ThisAnswVal
            $TakeAnswers="SELECT COUNT(id) AS _valcount FROM res_answers
WHERE idAnswer=".$ThisAnswVal." ";
            $res = mysql_query($TakeAnswers);
            if ($res){
                while($row = mysql_fetch_array($res)) {
                    $val_count=$row['_valcount'];
                }
            }
            else {
                echo "NO RESULTS FOUNDED FOR QUESTION ".($k+1). " <br>";
                exit();
            }
        }
    }
}

```

```
    }
    $QuestResults[] = $val_count;
    echo '<td align="right">'.$val_count.'</td>'; //значение кол-во ответов с ИД
$ThisAnswVal
    $QuestSumma = $QuestSumma + $val_count;
    }
    echo '</tr>';//конец строки
}
//закрываем таблицу
echo '</table>';
}
?>
```