Форма № Н-9.02

РІВНЕНСЬКИЙ ДЕРЖАВНИЙ ГУМАНІТАРНИЙ УНІВЕРСИТЕТ

Факультет математики та інформатики

Кафедра інформатики та прикладної математики

«До захисту допущено»
Завідувач кафедри
_____Батишкіна Ю.В.

Протокол №_____

від _____

Дипломний проект (робота)
ступеня ОКР Магістр

з спеціальності 122- Комп'ютерні науки

На тему: Інформаційна система методичної підтримки дисципліни «Веб- технології»

Виконав: здобувач освіти 2 курсу, групи КН-М-21

Тіт Назар Сергійович _____
(прізвище, ім'я, по батькові) (підпис)

Керівник: Кирик Т. А., ст. викладач _____
(посади, науковий ступінь, прізвище та ініціали) (підпис)

Рецензент: Турбал Ю.В., д. т. н, проф. каф комп. наук та прикл. матем. НУІВГП _____
(посади, науковий ступінь, прізвище та ініціали) (підпис)

Рецензент: Сяський В.А., к. т. н, доцент _____
(посади, науковий ступінь, прізвище та ініціали) (підпис)

Рівне – 2021 року

ЗМІСТ

ВСТУП

Актуальність теми

Протягом останніх років, відбувається процес переходу від традиційного навчання до дистанційного. Цей перехід відбувається через розвиток мережі Інтернет, це дає нам можливість пересилати необхідну кількість даних з одного кінця світу в інший, розміщувати інформацію на різноманітних ресурсах, роблячи її доступною. Сучасні інформаційні технології дають змогу підвищити та вдосконалити ефективність освітнього процесу.

В наш час, створення інформаційної системи підтримки дисципліни «Веб-технології» є необхідною. Однією з головних причин до упровадження веб-ресурсів є зручність та простота використання наявних інструментів для пошуку, та використання веб-ресурсів. Використовуючи ІС, можна суттєво підвищити ефективність навчального процесу. Студент має можливість знайти потрібну для себе інформацію, не зважаючи на те, в якому місці він буде перебувати. Також, щоб надати йому можливість набувати знання та перевіряти свої знання та навички дистанційно.

Об’єкт та предмет дослідження

Об’єкт дослідження – навчально-освітній процес. Предмет дослідження – інформаційна система підтримки дистанційного вивчення дисципліни «Веб-технології»

Мета і завдання дослідження

Мета та завдання дослідження в даній роботі – розглянути поняття інформаційної системи, дослідити основні властивості інформаційних систем та створити власну ІС для підтримки дисципліни «Веб-технології».

Методи дослідження

В даній роботі спроектовано веб-платформу за використанням методу моделювання.

Практичне значення отриманих результатів

Розроблено веб-платформу для підтримки дисципліни «Веб-технології».

Апробація

Виступ на звітній науковій конференції Рівненського державного гуманітарного університету за темою “Інформаційна система методичної підтримки дисципліни «Веб- технології»”

Структура роботи

Вступ, кількість розділів – 4, підрозділів – 13, висновки, список використаних джерел. Обсяг основної частини дослідження містить 63 сторінки, список використаних джерел містить 11 позицій.

РОЗДІЛ 1

ПОНЯТТЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ. ВИДИ ІС.

1.1. Поняття інформаційної системи

Інформаційна система – це сукупність організаційних і технічних засобів для збереження та обробки інформації з метою забезпечення інформаційних потреб користувачів.

ІС має декілька різних визначень терміну:

- Інформаційна система — є системою, комп'ютерною мережею або системою зв'язку
- Інформаційна система — організаційно-технічна система обробки інформації за допомогою технічних і програмних засобів
- Інформаційна система — система, призначена для одержання, обробки, зберігання, відображення та/або реєстрації даних про технічний стан конструкцій, систем, елементів, їх властивості та/або функціонування.

У багатьох випадках для того щоб створити власну інформаційну систему неможливо обійтися без використання БД (баз даних). Особливістю СУБД є забезпечення виконання запитів до БД. Ще одна важлива особливість у сучасніших СУБД виступає забезпечення режиму мультидоступу.

На сьогоднішній день розвинену архітектуру можна відносити до таких категорій: інформаційно-обчислювального потужного сервера, такий сервер містить в собі термінал або локальну розподілеу інформаційно-обчислювальну мережу серверів та робочих станцій, з цього випливає можливість спільного

використання ресурсів. Інформаційні системи можуть бути різними за своїми типами об'єктів, також вони поділяються за своїми характерами та розмірами задач, які потрібно виконати. Їх можна поділити за:

- ознакою розміщення, або напрямом діяльності, а саме державні, територіальні, галузеві, об'єднань, підприємств або установ;
- за своїм призначенням – пошуку інформації, довідки інформації, керуючі інформацією, системи підтримки прийняття рішень, інтелектуальні інформаційні системи;
- ступенем централізації обробки інформації - централізовані ІС, децентралізовані ІС, розподілені інформаційні системи;
- за ступенем інтеграції функцій - багато рівнів які мають інтеграцією за рівнями управління (підприємство - об'єднання, об'єднання - галузь), багаторівневі інформаційні системи, вони мають інтеграцією за рівнями свого планування;
- видом обробки інформації - фактографічні, документальні, документально-фактографічні, мультимедійні, текстові тощо;
- швидкістю обробки інформації - системи які існують в теперішньому часі, швиткої обробки транзакцій, пакетної обробки;
- видами діяльності - система автоматизованого проектування, автоматизовані ІС, автоматизовані системи управління технологічними процесами, корпоративні ІС.

Засобами розробки та впровадженням автоматизованих інформаційних систем включають в себе різновиди забезпечення які значно можуть допомогти у створенні та використанні. Такими видами забезпечення є технічні, програмні, інформаційні, організаційно-методичні, математичні, лінгвістичні, правові, технологічні.

1.2. Базові види забезпечення ІС

Базові види забезпечення ІС:

- *технічне* - сукупність технічних методів збору, пересилання, внесення, опрацювання, подачі і виводу інформації; комп'ютерна техніка та пристрої, носії інформації , монітори, клавіатури, прилади техніки організації, мережа тощо;
- *програмне* - операційні системи, інструментальні такі як редактори та таблиці електронного вигляду, прикладні;
- *інформаційне* - способи і пристрої заміни зовнішнього подання даних, описування відомостей в ході обробки, передачі інформації з машинного формату, відправлення в зовнішній через машинне (база даних, база знань, сховище даних, СУБД, файли) та немашинне забезпечення (методики, які описують принципи праці в інформаційних системах, системи класифікації та кодування, системи стандартизації документів тощо);
- *організаційно-методичне* - комплекс організаційно-методичних засобів, за допомогою яких описують або запускають працю технологію проектування, функціонування та розвитку ІС для вибраних її компонентів і видів забезпечень, які охоплюють методи і способи опису, формування, використання окремих організаційно-методичних процедур;
- *лінгвістичне* - комплекс мов програмування, які використовуються в ІС, мови управління і маніпулювання даними, методи систем пошуку, засоби проектування ІС, діалогові мови;
- *математичне* - сукупність засобів і методів, за допомогою яких будуються математичні моделі задач управління та алгоритм для того щоб їх вирішити;
- *правове* - комплекс норм, що представлені в нормативних документах, які встановлюють правовий статус ІС.

Кожна інформаційна система має власні виняткові характеристики та може класифікуватись як окремий об'єкт. Координація між типами інформаційних систем знаходиться в динаміці та постійному зростанні. У багатьох випадках ІС інтегруються у вигляді гібридних ІС, в яких циркулюють різні інформаційні потоки.

Застосунок інформаційної системи можна розглянути з різних сторін: характер діяльності, яких вони підтримують, функціонал, де їх використовують.

Вдале використання ІС потребує розуміння основних тенденцій розвитку, при цьому вони повинні забезпечуватися в основному:

a) облікові функції;
b) аналітично-звітні можливості;
c) можливості для роботи зі звітами;
d) можливість об'єднання даних;
e) можливості логічної, динамічної, статистичної та аналітичної обробки первинних даних;
f) надійність і безпечність ІС .

1.3. Історичні етапи розвитку інформаційних систем

В теперішньому часі швидко зростає потреба на інформацію та інформаційні послуги. В зв'язку з цим технології обробки інформації намагаються використовувати найбільш широкий спектр технічних засобів, комп’ютерну техніку та цифрові електронні засоби комунікації. На такій основі створюються обчислювальні системи та мережі не тільки для накопичення, збереження та перетворення інформації, а також для найбільшого приближення термінальних пристроїв до робочого місця фахівця або керівника, який приймає рішення.

Нинішні ІС виникли і функціонують завдяки таким технічним досягненням:

- швидким та містким засобам зберігання інформації (жорсткі та лазерні диски, флеш-пам'ять);
- цифровим пристроям зв'язку, які не створюють суттєвих обмежень на відстань та час (глобальні комп'ютерні мережі);
- апаратним та програмним пристроям автоматизованого опрацювання інформації (вибірка, сортування, подання в потрібній формі).

1. Перший етап є початковим етапом, він відбувається в 60-ті роки. Цей етап характеризується тим, що в ті роки відбувалось нагромадження опорного досвіду використання комп’ютерів, намагались виявити головні напрямки де вони можуть застосовуватись. Головною метою даного етапу є напрямок щоб зменшити управлінський апарат і кількість витрат на щоб його утримувати. На початку проводились роботи з автоматизації відокремлених операцій

бухгалтерського обліку, фінансових розрахунків, матеріально-технічного постачання.[**Error! Reference source not found.**]

2 .Другий етап - встановлення контролю над впровадженням нової інформаційної технології. Він відбувається в 70-ті роки. Цьому етапу властиві наступні ознаки:

- пошук напрямків де можливе використання комп'ютерів в управлінні;
- створення систем для організації управління комп'ютерною технікою, та виявити її вплив на процеси управління;
- ізольованість, як правило, несумісність окремих видів ІС організації;
- спрямованість на використання інформаційних технологій вузьким колом користувачів;
- створення вказівок, що до інтеграції інформаційного забезпечення управлінського персоналу;
- створення одної інформаційної служби, яка підпорядковується з адміністративних питань.[**Error! Reference source not found.**]

3. Інтеграція інформаційних систем (з 80-их років). Цьому етапу належать такі риси:

- на цьому етапі значно зменшується кількість технічних складнощів у галузі комп'ютерних технологій (відбувся великий прорив в розробці процесорів, оперативної пам'яті, створенні нові носії інформації) та комунікаційних засобів;
- впроваджуються потужні мережі, які починають об'єднуватись із системами комунікацій таких як: телефон, телетайп, радіо, телебачення.
- після того як з'явились персональні комп'ютери головним стає створення децентралізованих систем, до яких відносяться ПК, ЕОМ великої потужності, різне технічне забезпечення та різновиди обладнання об'єднуються в локальну мережу

- виконується потреба максимального зближення користувача та інформації, вона породжує створення в користувача враження, що потрібну інформацію користувач може знайти в себе на комп’ютері, хоча вона може знаходитись в різних вузлах локальної обчислювальної мережі;
- подається ідея “управління інформаційними ресурсами“, в даній концепції інформації розглядають, як окремий важливий ресурс, а саме фінанси, матеріали, обладнання і персонал;
- створюється новий зразок працівників, які вже звикли до того що використовуються нові інформаційні технології;
- відбуваються спроби на одній технічній, організаційній, методологічній основі впровадити автоматизовані ІС;
- підвищується значення інформаційних служб, в подальшому вони є службами управління ресурсами інформації в організації, за це відповідає віце-президент по інформації.[**Error! Reference source not found.**]

Основне завдання можна розглянути як комплекс наступних складових:

- збір інформації з різних джерел;
- реєстрація, опрацювання та подача інформації, яка характеризує стан виробництва та управління ним;

1.3.1. Види інформаційних систем

Розвиток комп'ютерної інформаційної технології невід'ємно пов'язаний з розвитком ІС, які використовуються для автоматизованого розв'язування задач. Для того щоб розв'язати будь-яку задачу за допомогою комп'ютера потрібно забезпечити розрахунки необхідними даними та створити математичну модель розв'язування задачі.

Завдання ІС — це створення інформації, яку потребує організація для забезпечення ефективного управління ресурсами , створення інформаційного середовища та технічного середовища за для управління організацією.

Інформаційна технологія — це спосіб перетворення інформації, він реалізується в межах ІС, де можуть використовуватися багато технологій. Проте вона глобальніша за ІС та може існувати поза нею.

Для вдалої роботи ІС потрібно наступне:

- виявлення інформаційних потреб;
- добір джерел інформації;
- збирання інформації;
- введення інформації із зовнішніх або внутрішніх джерел;
- опрацювання інформації;
- виведення інформації для надання її споживачам або передачі в іншу систему;
- організація використання інформації для оцінки тенденцій, розробки прогнозів, оцінки альтернатив рішень і дій, вироблення стратегії;
- організація зворотного зв'язку інформації, корекція вхідної інформації.

В основі систем є процес, зокрема в основі ІС — процес виробництва інформації. В такому розумінні ми можемо розглянути ІС як систему управління, де такий процес буде об'єктом управління.

За рівнем автоматизації ІС поділяються на три види:

- ручні – за таким рівнем інформаційні процеси будуть виконуватись людиною без застосування технічних засобів. На сьогодні дані ІС майже не використовуються;

- автоматизовані – у виконанні інформаційних процесів приймає участь як і людина так і технічні засоби (зарасз такі інформаційні системи є дуже поширені, в основному технічними засобами найчастіше є комп'ютери);
- автоматичні – реалізація інформаційних процесів відбувається без втручання людини. Людина бере участь у роботі такої ІС в етапі підготовки до роботи і етапі аналізу отриманих результатів).

За рівнем аналізу даних інформаційні системи також поділяютья на три види:

- системи опрацювання даних – такі ІС опрацьовують найлегші операції по обробці даних: упорядкування, перетворення, пошук, містять систему збереження та пошуку даних – базу даних, такі дані не мають аналізу;
- системи управління – такі ІС аналізують отримані дані, порівнюють їх із даними в плані, шукають певні потреби виробництва, відслідковують хід виконання проектів, встановлюють тенденції такі як закономірність та перспектива в роботі підприємств, організацій і цілих галузей господарства;
- системи підтримки прийняття рішень – такі ІС на основі аналізу отриманих даних узагальнюють їх та прогнозують майбутню діяльність організацій.

1.4. Система функцій управління

Майже всі різновиди ІС не зважаючи на сфери застосування включають один і той самий набір компонентів:

- функціональні компоненти;

- компоненти системи опрацювання даних;
- організаційні компоненти

При цьому під функцією управління можна розуміти як спеціальний постійний обов'язок однієї або декількох осіб, виконання якого повинне забезпечити досягнення певного результату.

Під функціональними компонентами потрібно сприймати систему функцій управління - повний комплекс пов'язаних між собою у часі й просторі роботи з управління, потрібних для досягнення поставлених перед підприємством задач. Тобто, будь-яка складна управлінська функція розділяється на ряд менших задач і в результаті доводиться до виконавця.

Наведені положення виділяються не тільки індивідуальним, а й груповим характером функції управління, а практичний результат створюється не епізодично, а постійно.

РОЗДІЛ 2. МОДЕЛІ ТА ТЕХНОЛОГІЇ ДИСТАНЦІЙНОГО НАВЧАННЯ.

2.1. Дистанційне навчання

У багатьох країнах світу форма дистанційного навчання доволі поширена. Поняття DISTANCE EDUCATION в світі відоме майже кожному і не потребує поясненню. Але для України така форма є новою формою навчання для здобуття освіти поряд з очною і заочною.

Дистанційна форма навчання – це отримання послуг освіти без відвідування навчальних закладів, а саме за допомогою сучасних інформаційно-освітніх технологій та системи телекомунікації, таких як електронна пошта, телебачення та Інтернет.

Дистанційне навчання можна впроваджувати у вищих навчальних закладах, а також і для того щоб підвищити кваліфікації й перепідготовки викладачів. Така форма навчання дозволяє отримати диплом тим, хто з різних причин не може навчатись очно. Отримавши навчальні матеріали в електронному чи друкованому вигляді і використанням телекомунікаційних мереж, учні можуть отримати знання та навички вдома, на робочому місці.

Якщо порівнювати з традиційними технологіями, технологія дистанційного навчання має багато вимог щодо повного методичного забезпечення учнів електронними підручниками і посібниками, системою навчальних завдань і тестових матеріалів контролю ефективності засвоювання знань.

Комп'ютерні системи можуть проекзаменувати, виявити помилки, дати необхідні поради, виконати практичне тренування, відкрити доступ

до бібліотек, за швидкий проміжок часу знайти потрібний розділ книги чи її параграф, абзац, цитату.

Навчальні курси можуть супроводжуватися ігровими ситуаціями, термінологічними словниками та відкривають доступ до основних вітчизняних і міжнародних БД і знань на будь-якій відстані та у будь-який час.

Під час дистанційного навчання важливим будуть індивідуальні можливості, потреби, темперамент і зайнятість студента. Він може проходити навчальні курси в різній послідовності, швидше чи повільніше. Ці фактори роблять дистанційне навчання якіснішим, доступнішим та дешевшим за традиційне.

Принципи дистанційного навчання:

- гнучкість – тривалість занять не регламентована;
- модульність – можливість коригування навчального плану до потреб студента;
- паралельність – навчання без відриву від виробництва;
- охоплення – залучення багатьох джерел інформації;
- технологічність – використання новітніх досягнень інформаційних та теле-комунікативних технології;
- залучення спеціалізованих форм контролю – дистанційно організовані іспити, співбесіди, екстернат, комп’ютерні тестові системи, прийом іспитів та заліків в режимі on-line.

Дистанційне навчання передбачає використання таких форм навчання:

- Лекції - на відміну від звичайних аудиторних лекцій, вони виключають живе спілкування з викладачем, але мають і ряд переваг. Щоб записати лекцію використовуються дискети, СД-диски тощо. Такі лекції можна відвідувати в будь який час і в будь-якому місці. Також, студентам не потрібно конспектувати матеріал лекцій.

- Лабораторні роботи – такі лабораторні мають призначення щоб засвоїти практичний матеріал . У традиційній системі освіти лабораторні роботи потребують спеціального обладнання. Можливості дистанційного навчання можуть сильно спростити проведення таких робіт за допомогою використання мультимедіа-технологій, імітаційного моделювання тощо. Віртуальна реальність дає можливість побачити студентам явища як в звичайних умовах показати тяжко або інколи і неможливо.

- Контрольні роботи - це система індивідуальних, частіше тестових завдань, зосереджених на перевірку теоретичного результату та практичного результату засвоєння студентами навчального матеріалу.

- Консультації - є однією з форм керівництва роботою студентів і надає їм допомогу в самостійному вивченні дисципліни. Частіше за все, для цього використовується телефон і електронна пошта. Надається можливість організовувати консультації навчальної аудиторії викладачем у режимі онлайн. При цьому кожен учень у певний, попередньо зазначений час за допомогою доступу в мережу Інтернет має можливість ставити викладачеві запитання, пов'язані з вивченням певного предмета, та отримати на них кваліфіковані відповіді. Крім того, консультації дають змогу педагогові краще

оцінити особистісні якості студента, який навчається, і його інтелект.

2.2. Моделі дистанційного навчання

Дистанційне навчання базується на принципах самостійності, науковості, системності та систематичності. Також важливо забезпечувати принципи активності у навчання, постійно підтримувати зв'язок теорії з практикою. Важливими також є демократизація навчання, оптимізація процесу навчання, принципи інтерактивності, диференціації та не традиційності системи навчання.

Одним із методів наукового дослідження є метод моделювання. Цей метод заснований на створенні структури організації та умов функціонування певного процесу, навчальної діяльності у тому числі. Використання моделювання в процесі навчання є допоміжним засобом для здійснення аналізу та оцінки навчального процесу[3].

Моделювання є допоміжним засобом планування всього процесу, та допомагає передбачити результат. Метод моделювання ґарантує дотримання принципів системності й послідовності[8]. Розглянемо моделі організації ефективного навчання з використанням ІКТ:

1. Змішана модель навчання об'єднує у собі очну та дистанційну форми. Дану модель дослідники пропонують використовувати у сучасних умовах. Модель застосовна у середніх навчальних та вищих навчальних закладах. Викладач може продумати модель навчання, визначити види діяльності, які плануватимуться очно (оглядові лекції, семінарські й практичні заняття, заліки, іспити, захисти курсових, бакалаврських, магістерських робіт) та дистанційно. При такій формі навчання викладач

постійно повинен заохочувати, допомагати, спостерігати за перебігом навчання. Також викладач має можливість редагувати процес під потреби кожного студента.

2. Модель мережевого навчання(за допомогою Інтернет) придатна в умовах, коли немає можливості організувати очне чи заочне навчання. Така форма навчання є оптимальною серед тих, працює та паралельно навчається. працюючих студентів, які окрім того працюють і хочуть паралельно навчаються, для самостійного вивчення певних курсів , осіб з обмеженими фізичними можливостями, учнів, студентів в умовах пандемії. Використання такого навчання потребує детальної розробки дистанційних курсів. В. Кухаренко пропонує трьох рівневу систему запровадження дистанційного навчання. Під час першого етапу студенти вивчатимуть дисципліни із завданнями, які повторюватимуться та відповіді для яких визначено. На другому етапі викладач взаємодіє зі студентом, спрямовує його навчання. На третьому етапі пропонується участь у навчальному процесі провідних вчених у конкретних галузях із використанням новітніх засобів комунікації . В умовах пандемії така форма навчання дозволяє продовжувати навчальний процес. Якість такого навчання здебільшого нижча, ніж у першому випадку[6].

Інтерактивне телебачення, відео-конференції – у такій моделі навчання можливе тільки за допомогою використання телевізійного обладнання та відеокамер. В Україні запроваджено теле-уроки для школярів. Така форма навчання потребує великих фінансових витрат, але дає змогу імітувати очну форму навчання. Використання відео-зв’язку викладачем дозволяє організувати контакт між викладачем та студентами, також відкриває додаткові можливості спілкування між студентами при вирішення робочих завдань. Моделювання є головним фактором ефективності у всіх формах навчального

процесу. На думку вчених, характеристики моделі дистанційної форми навчання, в основі якої лежить загально-теоретична засада моделювання: так модель повинна містити компоненти, які мають вплив на якість навчального процесу; в моделі повинна бути структуру, яку можна легко діагностувати та контролювати на всіх рівнях її реалізації; модель повинна не тільки контролювати освітній процес, а й мати можливість його змінювати[5].

У системі дистанційної освіти можна виділити такі компоненти: цілі, які визначені соціальним замовленням для кожної з форм навчання; зміст, який буде залежати від навчальних програм закладу освіти, методи, організаційні форми, засоби навчання .

Особливістю дистанційного навчання є спілкування студента, який навчається, з іншими студентами і педагогами. Вони всі об’єднані однією метою. Для організації дистанційного навчання потрібно щоб всі учасники взаємодіяли між собою:

Студент – викладач. Постійна взаємодія студента з викладачем є визначальною , особливо для студентів які навчаються на 1–2 курсі і які не можуть самостійно організовуватись. Дистанційне навчання є формою самоосвіти, у ній велику роль відіграє педагог: моделює процес навчання, збільшує мотивацію, підбирає та визначення форми подачі матеріалу, методи навчання, контролює процес. Викладач підтримує та забезпечує зворотний зв’язок.

1. Студент – студент. Під час такої взаємодії відбувається обмін досвідом що до виконання завдань, вивчення нових тем, пошуку матеріалів. Спілкування відбуватиметься в відео конференції, соціальних мережах.

Результатом буде формування навичок співпраці, ці навички будуть потрібними у майбутньому. Так як людина – елемент соціуму, то така взаємодія буде покращуватись емоційний стан, та полегшить сприйняття дистанційного навчання.

2. Студент – адміністрація. Адміністрація ЗВО організовує навчальний процес (розклад занять), матеріально-технічне забезпечення, підтримку з використання ІКТ, створення інформаційно-освітнього простору, гарантування дотримання прав, контроль за виконанням обов'язків.
3. Викладач – адміністрація. В кінці такої співпраці є розроблені та затверджені навчальні програми, плани, розклад занять, створене інформаційне середовище, система управління, комфортні умови як для студентів, так і для науково-педагогічного складу. Взаємодія викладачів й адміністрації сприятиме швидкому вирішенню проблем.

Ефективними способами комунікації всіх учасників процесу є спілкування під час відео конференції, аудіо конференції, на різних навчальних платформах, у соціальних мережах, за допомогою електронної пошти. Підвищує мотивацію та позитивно впливає на процес навчання онлайн-спілкування в режимі реального часу.

4. Діагностично-оцінювальний. Це є складник освітнього процесу. На цьому етапі перевіряють знання студентів, оцінюють, аналізують педагогічний процес.

До активнішої діяльності здобувачів освіти мотивують створення ситуації успіху, позитивне оцінювання, відзначення здобутків. Визначення нових цілей, демонстрація перспектив навчальної діяльності, можливостей практичного використання здобутих знань, умінь, навичок вироблятимуть потребу в безперервному навчанні.

5. Соціально-емоційний. Головним недоліком дистанційного навчання є обмеження спілкування, взаємодії з іншими учасниками процесу. Також пандемія, невизначене майбутнє, зменшення соціальних контактів негативно впливають на психологічний та емоційний стан студентів. Через це викладачі, адміністрація навчального закладу повинні потурбуватись про доброзичливе ставлення до кожної людини, використовуючи ненасильницьку комунікацію та знаходячи спільну мову. Лише в безпечних, комфортних умовах студент може зосереджуватись на навчанні, може мислити, експериментувати, творити, винаходити. Важливо бачити реакцію студентів на зміст навчання, практичну діяльність яка пропонується, аналізувати активність під час виконання поставлених завдань для того, щоб вирішити потреби майбутніх фахівців.

Ефективність навчання може підвищити конструктивний зворотний зв’язок, який є потрібний так як і для педагога, так і для студентів: викладачеві допоможе вдосконалити курс та організацію навчальної діяльності, студентам – повірити у свої можливості, спонукатиме їх активніше працювати. Завдяки зворотному зв’язку студенти будуть розуміти на скільки їхня думка є важливою, мають відповідальність за навчальний процес, а також мають змогу змінювати освіту в кращу сторону відповідно до потреб сучасності. Перерахуємо ознаки які визначають ефективність навчальної моделі:

- результативність;
- універсальність;
- оптимальність;
- гнучкість.

РОЗДІЛ 3. «ВЕБ-ТЕХНОЛОГІЇ». ОСОБЛИВОСТІ ВИКЛАДАННЯ ДИСЦИПЛІНИ В УМОВАХ ДИСТАНЦІЙНОГО НАВЧАННЯ.

3.1. Поняття «Веб-технології»

Веб-технологія відноситься до різних інструментів і методів, які використовуються в процесі комунікації між різними типами пристроїв через Інтернет. Для доступу до веб-сторінок використовується веб-браузер. Веб-браузери можна визначити як програми, які відображають текст, дані, зображення, анімацію та відео в Інтернеті. Доступ до ресурсів із гіперпосиланнями у всесвітній павутині можна отримати за допомогою програмних інтерфейсів, наданих веб-браузерами.

Існує декілька технологій створення веб-сторінок. Їхню класифікацію можна пояснити таким чином:

Технології для розробки статичних сайтів передбачають собою використання мови розмітки гіпертексту HTML та каскадних таблиць стилів CSS Cascading Style Sheets.

Завдяки мови HTML можна змінювати текст, відокремлювати в ньому елементи функціональну, а також робити гіперпосилання і додавати на сторінку зображення, звук та різні мультимедійні елементи.

На сьогодні, окрім HTML використовуються і інші мови розмітки, а саме WML, XML.

CSS використовують щоб мати можливість оформлювати та редагувати елементи у веб-сторіноках, в кінцевому результаті значно знижуючи масштаб веб-сторінок.

Створення таких веб-сторінок як статичний сайт - це процес який потребує багато праці. Таку сторінку потрібно створювати вручну у файловій системі комп'ютера, потім її потрібно зберегти і завантажити на сайт.

Під час розробки такої сторінки використовують два основні підходи:

1) першим підходом є ручне створення, яке відбувається за допомогою невізуальних редакторів тексту типу. Це дає можливість використовувати додаткові функції для зручності, а саме підсвітка коду, перевірка валідності створеного коду, вибір кольору, швидке створення таблиць, вставку основних й найрозповсюджених елементів гіпертекстової розмітки.

2) користування візуальними редакторами такого типу як Microsoft FrontPage, Macromedia Dreamweaver. Зазвичай одним великим мінусом є - створення великого програмного коду, в результаті якого збільшується вага сторінки та швидкість її загрузки. По такій причині більшу перевагу матимуть невізуальні редактори.

Всі зміни на веб-сторінці відбуваються через виправлення HTML-коду. Кожного разу, коли буде потреба змінити щось на такій сторінці, користувач зможе покращити на своєму комп'ютері, спочатку зберігаючи, а після повторно завантажує на сайт. Такі дії може виконати тільки адміністратором сайту.

Сучасними технологіями для створення динамічних сайтів передбачається використання програмного коду, це забезпечить інтерактивність Web-сторінок, і буде називатися скриптом.

Скрипт - це програма, яка автоматизує деяку задачу.

Скри́птова мова (*scripting language*, мова сценаріїв) — мова програмування, розроблена для запису «сценаріїв», послідовностей операцій, які користувач може виконувати на комп'ютері. Сценарії, зазвичай, інтерпретуються, а не компілюються.

Розрізняють сценарії, які виконуються на стороні клієнта і такі, що виконуються на стороні сервера. Сценарії які використовують на стороні клієнта називаються клієнтські скрипти, вони знаходяться під керуванням браузера, на стороні сервера мають назву серверні скрипти - керуються Web-сервером.

Клієнтські скрипти можна використовувати, коли при внесенні інформації перед відправкою її на сервер, для початку відбувається перевірка її коректності за допомогою скрипта. Такий спосіб значно зменшує навантаження на сервер і створює більшу зручність для користувача.

Перевагою клієнтських скриптів є те, що вони можуть перевіряти правильність інформації, яку вносить користувач, і обробляти її, не звертаючись до сервера.

Недоліками клієнтських скриптів є :

- За допомогою клієнтського скрипта нічого не можна записати на сервер. За допомогою такого скрипта неможливо зробити, до прикладу, гостьову книгу.
- Опрацювання скрипта залежить від самого браузера який використовує користувач. Він матиме властивість налаштувати свій браузер так, щоб він повністю ігнорував написані скрипти.

- Код клієнтського скрипта може подивитися всі користувачі, хто вибравши в меню браузера команду "Вихідний код сторінки", після чого відкриють сторінку зі скриптом.

Зазвичай клієнтські скрипти, пишуться на мовах JavaScript, Java. Можуть використовуватися VBScript, ActionScript (Flash) та інші мови програмування. Існують випадки, коли текст скрипта вбудований в html-документ, у всіх інших випадках він завантажується з окремого файлу.

За допомогою JavaScript, найчастіше, виконуються такі ефекти, як, поява спливаючих вікон з повідомленнями, відображення анімації. Крім того, JavaScript-сценарії часто використовуються для визначення типу браузера і платформи, на якій він виконується. JavaScript-сценарії також успішно застосовуються для перевірки коректності даних, введених користувачем.

У VBScript є багато однакового з мовою JavaScript, але він працює тільки з Microsoft Internet Explorer, це обмежує сферу її використання.

У документ HTML можна також вмістити флеш-фрагменти або swf-файли. Флеш створює інтерактивність за рахунок інтерактивної векторної анімації для Web. Щоб створити Флеш, потрібно використовувати мову сценаріїв ActionScript.

Окрім цього, почали набирати популярності такі технології як AJAX, Adobe Flash, Microsoft Silverlight. Також можуть застосовуватися Java-аплети і технологія ActiveX.

У разі того , що скрипти, які являються клієнтськими, збільшують розміри веб-сторінок, їх кількість і обсяг на сторінці повинен бути обмеженими.

Створення статичних сайтів з інтерактивними веб-сторінками доцільно виконувати в редакторі Macromedia Dreamweaver.

Серверні скрипти виконуються на стороні сервера під керуванням веб-сервера.

Коли користувач створює запит на довільну сторінку , переходить на неї за посиланням або вводить адресу в адресному рядку свого браузера, таким чином потрібна сторінка спочатку обробляється на сервері, тобто виконуються всі скрипти, які пов'язані із формуванням певної сторінки, і тільки після цього вона вертається до користувача у вигляді HTML-документа . Користувач не має можливості побачити код серверного скрипта.

Серверні скрипти потрібні, у випадку, коли web-сервер повинен повернути інформацію для різних користувачів, а також інформацію, яка змінюється з перебігом часу (найбільш актуальним прикладом є форум).

Серверні скрипти, частіше за все, взаємодіють з базами даних. У такому разі скрипти будуть виконувати запити до СУБД, останнє повертатиме результати, а скрипт створює документ, який відправляється до браузеру. Найпопулярнішими є СУБД Mysql, MS SQL Server, Oracle.

Серверні скрипти напряму залежить від сервера, на якому розміщений сайт, а також, від технологій якими підтримується сервер.

В залежності які задачі повинні виконуватись для того щоб створити сайт вибирають мову серверних скриптів. Щоб створити інтерактивні сайти різного розміру, правильним буде використати мову сценаріїв PHP.

Головним плюсом мови PHP буде те, що вона є безкоштовною, а також має відкриті вихідні коди і працює на багатьох платформах.

Існуючі сайти зазвичай представляють з себе складні програмно-інформаційні системи, вони потрібні для вирішення багатьох задач, таких як бізнес-логіка і за своєю теорією можуть повторювати функції більшості бізнес-застосувань.

Щоб підтримувати роботу таких сайтів потрібно, щоб сервера було спеціалізоване програмне забезпечення, яке буде працювти з базою даних і створюватиме сторінки динамічно. Дане програмне забезпечення має назву веб-застосування.

Веб-застосування — це програмний комплекс для вирішення задач веб-сайту, який міститься у складі сайту і працює разом з ним.

Зазвичай, веб-застосування створюють як доповнення в такій архітектурі як "клієнт-сервер": клієнтська частина реалізує користувальницький інтерфейс, створює запити до сервера і опрацьовує відповіді від нього; клієнт задає запит в серверну частину, в результаті чого виконується розрахунок, далі формує веб-сторінку і надсилає її клієнтові по мережі з використанням протоколу HTTP.

Клієнтом веб-застосування є браузер, він частіше за все містить деяку частину логіки застосування, а саме перевірки коректності даних, що вводяться, а сервером — веб-сервер. При цьому серверна частина містить різні архітектурні рішення.

3.1.1. Всесвітня павутина (WWW)

World Wide Web скорочено WWW і широко відомий як в Інтернеті. WWW була ініційована CERN (Європейська бібліотека ядерних досліджень) у 1989 році.

Це проект, створений Лі Тімоті Бернером у 1989 році для ефективної спільної роботи дослідників у ЦЕРН. це організація під назвою World Wide Web Consortium (W3C), яка була розроблена для подальшого розвитку в Інтернеті. Режисером цієї організації є Лі Тім Бернер, він же батько Інтернету. [**Error! Reference source not found.**]

З точки зору користувача, Інтернет складається з величезного, всесвітнього зв'язку документів або веб-сторінок. Кожна сторінка може містити посилання на інші сторінки в будь-якій точці світу. Сторінки можна отримати та переглянути за допомогою браузерів, серед яких Internet Explorer, Netscape Navigator, Google, Chrome тощо є популярними. Браузер отримує запитану сторінку, інтерпретує текст і команди форматування на ній, а також відображає сторінку, правильно відформатовану, на екрані.

Основна модель роботи мережі показана на малюнку нижче. Тут браузер відображає веб-сторінку на клієнтській машині. Коли користувач натискає рядок тексту, який пов'язаний зі сторінкою на сервері abd.com, браузер переходить за гіперпосиланням, надсилаючи на сервер abd.com повідомлення із запитом на сторінку.

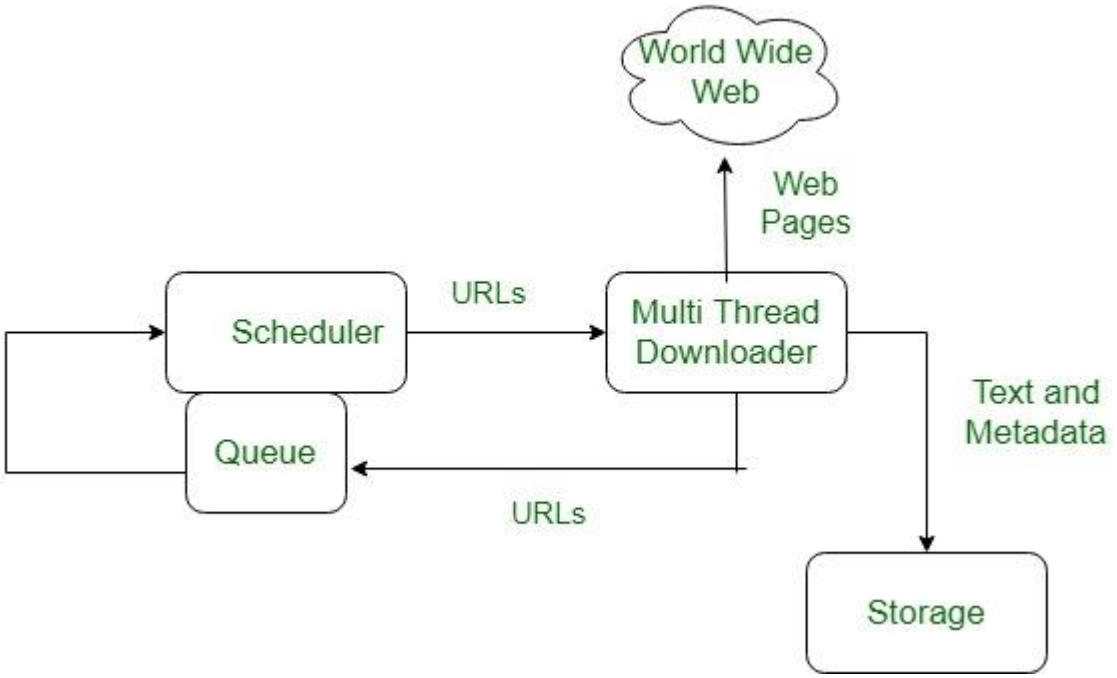

Рис. 3.1. Модель роботи мережі

Тут браузер відображає веб-сторінку на клієнтській машині, коли користувач натискає рядок тексту, який пов’язаний зі сторінкою на abd.com, браузер переходить за гіперпосиланням, надсилаючи повідомлення серверу abd.com із запитом на сторінку.

Всесвітня павутина заснована на кількох різних технологіях: веб-браузерах, мові розмітки гіпертексту (HTML) і протоколі передачі гіпертексту (HTTP).

Для доступу до веб-сторінок використовується веб-браузер. Веб-браузери можна визначити як програми, які відображають текст, дані, зображення, анімацію та відео в Інтернеті. Доступ до ресурсів із гіперпосиланнями у всесвітній павутині можна отримати за допомогою програмного інтерфейсу, наданого веб-браузерами. Спочатку веб-браузери використовувалися лише для серфінгу в Інтернеті, але тепер вони стали більш універсальними. Веб-браузери можна використовувати для виконання кількох завдань, включаючи пошук,

розсилку, передачу файлів та багато іншого. Деякі з часто використовуваних браузерів - Internet Explorer, Opera Mini, Google Chrome.

Особливості WWW**:**

- Інформаційна система гіпертексту
- Кроссплатформенність
- Поширений
- Відкриті стандарти та відкритий код
- Використовує веб-браузери для забезпечення єдиного інтерфейсу для багатьох служб
- Динамічний, інтерактивний та розвивається.
- «Веб2.0»

Компоненти Інтернету**:** Існують 3 компоненти Інтернету:

1. Уніфікований локатор ресурсів (URL): служить системою для ресурсів у мережі.
2. HyperText Transfer Protocol (HTTP): визначає зв’язок браузера та сервера.
3. Мова розмітки гіпертексту (HTML): визначає структуру, організацію та вміст веб-сторінки.

3.1.2. Веб-браузер

Коли нам потрібна яка-небудь інформація, більшу частину часу ми отримуємо допомогу з Інтернету, і ми отримуємо інформацію. Інтернет легко надає нам корисну інформацію; ми використовуємо мобільні телефони, комп'ютери і планшети. Ми шукали багато речей в нашому повсякденному житті, тому ми отримуємо інформацію про усьому світі, але ми не можемо

отримати інформацію, просто підключившись до Інтернету. Нам потрібна платформа, на якій ми могли б шукати відповіді на наші запитання. Платформа, що надає такого роду послуги, називається веб-браузером, без веб-браузера інтернет не зможе надавати інформацію.

Що таке веб-браузер?

Веб - браузер-це прикладне програмне забезпечення для вивчення www (Всесвітньої павутини). Він забезпечує інтерфейс між сервером і клієнтом і запитує у сервера веб-документи і служби. Він працює як компілятор для візуалізації HTML, який використовується для створення веб-сторінки. Всякий раз, коли ми шукаємо щось в Інтернеті, браузер завантажує веб-сторінку, написану у форматі HTML, включаючи текст, посилання, зображення та інші елементи, такі як таблиці стилів і функції JavaScript. Google Chrome, Microsoft Edge, Mozilla Firefox, Safari є прикладами веб-браузерів.

Історія веб - браузера

Перший веб-браузер WorldWideWeb був винайдений в 1990 році Тімом Бернерсом-Лі. Пізніше це стає Сполучною ланкою. В 1993 році Марк Андресс і їх команда винайшли нову браузерну мозаїку. Це був перший браузер, який одночасно відображав текст і зображення на екрані пристрою. Він також винайшов ще один браузер Netscape в 1994 році. В наступному році Microsoft запустила веб-браузер Internet Explorer, який вже був встановлений в операційній системі Windows. Після цього було винайдено безліч браузерів з різними функціями, такими як Mozilla Firefox, Google Chrome, Safari, Opera і т. д.

Як працює веб - браузер?

Веб-браузер допомагає нам знаходити інформацію в будь-якому місці Інтернету. Він встановлюється на клієнтський комп'ютер і запитує інформацію з веб-сервера. Такий тип робочої моделі називається клієнт-серверної моделлю.

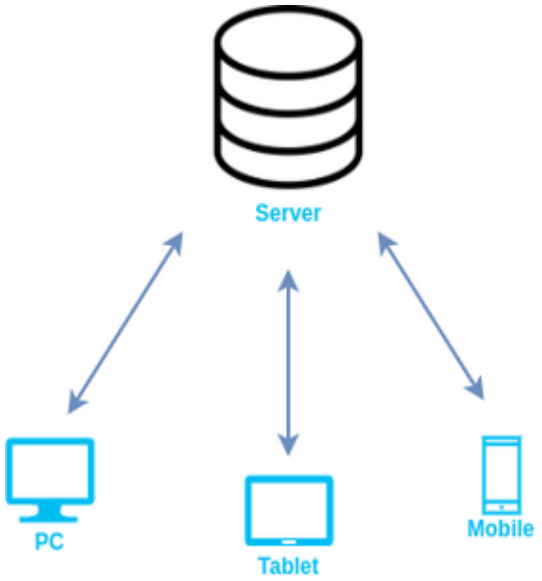

Рис. 3.2. Вигляд клієнт-серверної моделі

Файли cookie Веб-сайту

Коли ми відвідуємо який-небудь веб-сайт через Інтернет, наш веб-браузер зберігає інформацію про нас в невеликих файлах, які називаються файлами cookie. Файли cookie призначені для запам'ятовування інформації про стан нашої історії відвідувань. Ще деякі файли cookie використовуються для запам'ятовування про нас, таких, як наші інтереси, наші шаблони перегляду. Веб-сайти показують нам рекламу, засновану на наших інтересах, з використанням файлів cookie.

Деякі популярні веб-браузери

Існують деякі популярні і найбільш часто використовувані веб-браузери, такі як Google Chrome, Mozilla Firefox, Microsoft Edge, Safari.

Google Chrome-самий популярний веб-браузер в світі. У 77% пристроїв використовується Google Chrome. Цей браузер був розроблений компанією Google в 2008 році для Microsoft Windows. Пізніше він використовувався в операційних системах macOS, Linux, Android, iOS. Це дуже надійний браузер, доступний на 47 мовах. Процес установки Google Chrome дуже простий і безкоштовний для всіх.

Mozilla Firefox також відомий як браузер Firefox, розроблений Mozilla Foundation та корпорацією Mozilla в 2002 році. Він доступний в операційних системах Linux, Microsoft Windows, Android і iOS. В системі Linux Mozilla Firefox є браузером, встановленим за замовчуванням.

3.1.3. Веб-сервер та його типи

Веб-сервер: Веб-сервер-це програма, яка обробляє мережеві запити користувачів, і обслуговує їх файлами, створюють веб-сторінки. Цей обмін відбувається з використанням протоколу передачі гіпертексту (HTTP). По суті, веб-сервери-це комп'ютери, які використовуються для зберігання HTTP-файлів, які створюють веб-сайт, і коли клієнт запитує певний веб-сайт, він доставляє запитаний веб-сайт клієнту. Наприклад, ви хочете відкрити Facebook на своєму ноутбуці і ввести URL-адресу в рядку пошуку Google. Тепер ноутбук відправить HTTP-запит для перегляду веб-сторінки facebook на інший комп'ютер, відомий як веб-сервер. Цей комп'ютер (веб-сервер) містить всі файли (зазвичай у форматі HTTP), які складають веб-сайт, такі як текст, зображення, gif-файли і т. д.Після обробки запиту веб-сервер відправить запитані файли, пов'язані з веб-сайтом, на ваш комп'ютер, і потім ви зможете перейти на веб-сайт. Різні веб-сайти можуть зберігатися на одних і тих же або різних веб-серверах, але

це не впливає на фактичний веб-сайт, який ви бачите на своєму комп'ютері. Веб - сервер може бути будь-яким програмним або апаратним забезпеченням, але зазвичай це програмне забезпечення, що працює на комп'ютері. Один веб-сервер може обробляти декілька користувачів у будь-який момент часу, що є необхідністю, в іншому випадку для кожного користувача повинен був бути веб-сервер, і, враховуючи нинішнє населення світу, це майже неможливо.Веб-сервер ніколи не відключається від Інтернету, тому що якщо б це було так, то він не зміг би отримувати будь-які запити і, отже, не міг би їх обробляти.

На ринку є безліч веб-серверів, як безкоштовних, так і платних. Деякі з них описані нижче:

- HTTP-сервер Apache: Це самий популярний веб-сервер, і близько 60 відсотків комп'ютерів з веб-серверами в світі використовують цей веб-сервер. Веб-сервер Apache HTTP був розроблений Apache Software Foundation. Це програмне забезпечення з відкритим вихідним кодом, що означає, що ми можемо отримати доступ до його кодом, вносити в нього зміни і змінювати його відповідно з нашими уподобаннями. Веб-сервер Apache можна легко встановити і використовувати практично у всіх операційних системах, таких як Linux, macOS, Windows і т. д.
- Інформаційні служби Microsoft Internet (IIS): IIS (інформаційні служби Інтернету) - це високопродуктивний веб-сервер, розроблений корпорацією Microsoft. Він тісно пов'язаний з операційною системою і тому відносно простий в адмініструванні. Він розроблений корпорацією Майкрософт, в ньому є хороша система підтримки клієнтів, до якої легше отримати доступ, якщо ми зіткнемося з якою-небудь проблемою з сервером. Він володіє всіма функціями HTTP-сервера Apache, за винятком того, що він не є програмним забезпеченням з відкритим вихідним кодом, і тому його код недоступний, що означає, що ми не можемо вносити зміни в код згідно з нашими

потребами. Він може бути легко встановлений на будь-якому пристрої Windows.

- Lighttpd: Lighttpd вимовляється як "Злегка". В даний час він обслуговує близько 0,1 відсотка веб-сайтів у світі. Lighttpd має невелике навантаження на процесор і тому порівняно легше запускається. Він займає мало місця в пам'яті і, отже, порівняно з іншими веб-серверами, вимагає менше місця для роботи, що завжди є перевагою. Він також має оптимізацію швидкості, що означає, що ми можемо оптимізувати або змінювати його швидкість у відповідності з нашими вимогами. Це програмне забезпечення з відкритим вихідним кодом, що означає, що ми можемо отримати доступ до його коду і вносити в нього зміни у відповідності з нашими потребами, а потім завантажити наш власний модуль (змінений код).
- Сервер Jigsaw: Jigsaw написаний на мові Java і може запускати сценарії CGI (common gateway interference), а також програми на PHP. Це не повноцінний сервер, і він був розроблений як експериментальний сервер для демонстрації нових веб-протоколів. Це програмне забезпечення з відкритим вихідним кодом, що означає, що ми можемо отримати доступ до його коду і вносити в нього зміни у відповідності з нашими потребами, а потім завантажити наш власний модуль (змінений код). Він може бути встановлений на будь-якому пристрої за умови, що пристрій підтримує мову Java і модифікації в Java.
- Система Sun Java: Система Sun Java підтримує різні мови, сценарії та технології, необхідні для Web 2.0, такі як Python, PHP і т. д. Це не програмне забезпечення з відкритим вихідним кодом, і тому його код недоступний, що означає, що ми не можемо вносити зміни в код згідно з нашими потребами.

3.2. Web Development

Веб-розробка відноситься до створення, створення і обслуговування веб-сайтів.Вона включає в себе такі аспекти, як веб-дизайн, веб-публікація, веб-програмування і управління базами даних.Це створення програми, що працює через Інтернет, то є веб-сайти.

Слово "Web Development" складається з двох слів, тобто:

- Web **(Мережа):** Це відноситься до веб-сайтів, веб-сторінок або чого-небудь, що працює через Інтернет.
- Development **(Розвиток):** Створення програми з нуля.

Веб-розробку можна розділити на два види:
Frontend Development – розробка інтерфейсу
Backend Development – розробка бекенду

3.2.1. Розробка Інтерфейсу

Розробка Інтерфейсу: Частина веб-сайту, з якої користувач взаємодіє безпосередньо, називається інтерфейсом. Це також називається "клієнтської стороною" програми.

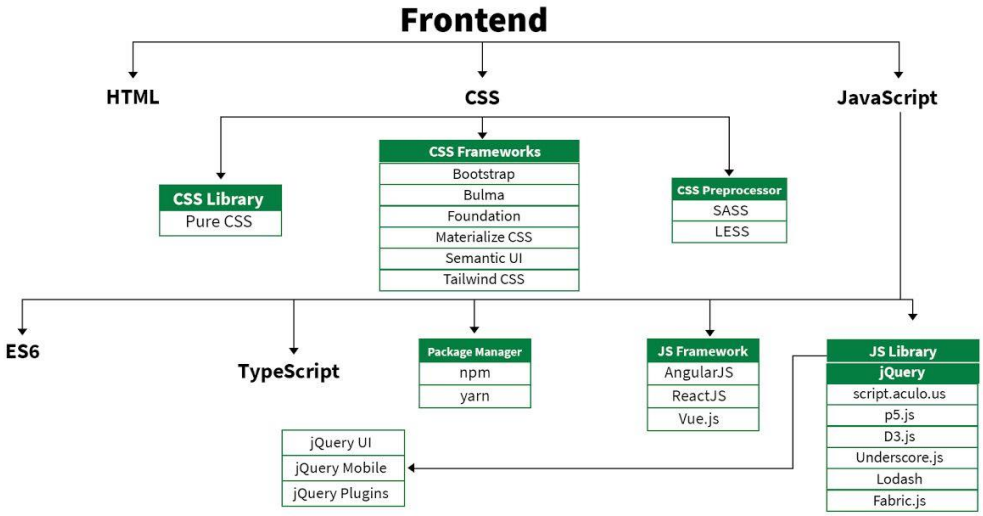

Рис. 3.3. Структура інтерфейсу

- HTML: HTML розшифровується як мова розмітки гіпертексту.Він використовується для розробки інтерфейсної частини веб - сторінок з використанням мови розмітки.Він діє як каркас для веб-сайту, оскільки використовується для створення структури веб-сайту.
- CSS: Каскадні таблиці стилів, які з любов'ю називають CSS, - це проста мова, призначений для спрощення процесу створення презентабельних веб-сторінок.Він використовується для оформлення нашого веб-сайту.
- мова JavaScript: JavaScript-це мова сценаріїв, яка використовується для забезпечення динамічної поведінки нашого веб-сайту.
- Bootstrap: Bootstrap-це безкоштовна колекція інструментів з відкритим вихідним кодом для створення адаптивних веб-сайтів і веб-додатків.Це найпопулярніший CSS-фреймворк для розробки адаптивних мобільних веб-сайтів.В даний час веб-сайти ідеально підходять для всіх браузерів (IE, Firefox і Chrome) і для екранів всіх розмірів (настільні комп'ютери, планшети, фаблеты і телефони).

Інтерфейсні фреймворки та бібліотеки:

- AngularJS
- React.js
- VueJS
- jQuery
- Bootstrap
- Material UI
- Tailwind CSS
- jQuery UI

Що робить інтерфейсний розробник? Отже, інтерфейсний розробник відповідає за розробку користувальницького інтерфейсу веб-сайту. В цілому, інтерфейсні розробники працюють над аспектами дизайну і верстки веб-сайтів так само, як і внутрішні розробники, які відповідають за процеси на стороні сервера, такі як управління базами даних, інтеграція API і т. д. Інтерфейсний розробник також займається впровадженням візуальних елементів, які можуть підвищити продуктивність веб-сайту і забезпечити кращий користувальницький інтерфейс. Деякі з основних ***ролі та обов'язки*** інтерфейсних розробників перераховані нижче:

- Реалізація веб-дизайну і структури
- Приходьте з різними ідеями для поліпшення користувацького досвіду
- Гарантує, що веб-дизайн буде адаптивним, безпечним і масштабованим
- Будьте в курсі останніх тенденцій веб-дизайну і т. д.

3.2.2. Backend Development

Backend Development**:** Backend Development-це серверна частина веб-сайту. Це та частина веб-сайту, яку користувачі не можуть бачити і взаємодіяти

з нею. Це частина програмного забезпечення, яка не вступає в прямий контакт з користувачами. Він використовується для зберігання та впорядкування даних.

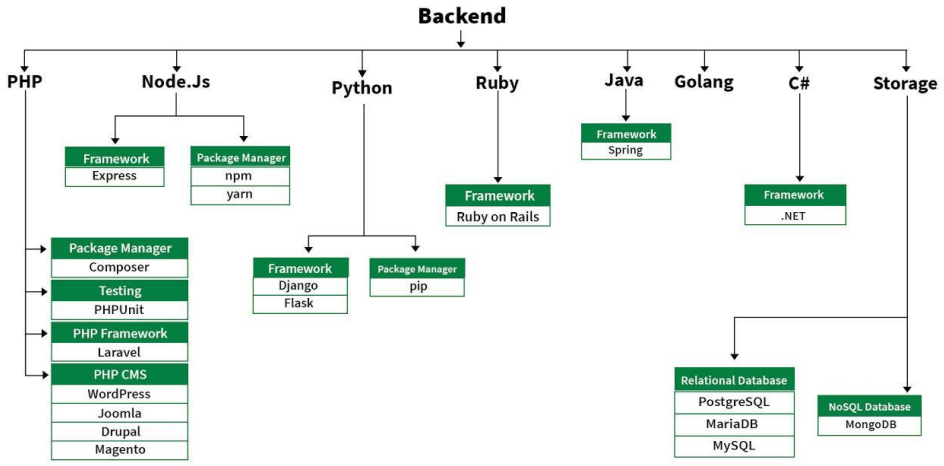

Рис. 3.4. Вигляд структури серверної частини

- PHP: PHP-це серверний мова сценаріїв, розроблений спеціально для веб-розробки.
- Java: Java-один з найбільш популярних і широко використовуваних мов програмування. Він володіє високою масштабованістю.
- Python: Python-це мова програмування, який дозволяє швидко працювати і більш ефективно інтегрувати системи.
- Node.js: Node.js це платформна середовище виконання з відкритим вихідним кодом для виконання коду JavaScript поза браузера.
- Back End Frameworks: Express, Django, Rails, Laravel, Spring

3.3. Дисципліна «Веб-технології»

Метою навчальної дисципліни є засвоєння необхідних знань з основ веб технологій та веб-дизайну, а також формування твердих практичних навичок

щодо розробки якісних сайтів. "Веб-технології та веб-дизайн" – навчальна дисципліна, що вивчає інструментарій розробки веб-сайтів з використанням веб-програмування та вебдизайну. Об'єктом навчальної дисципліни є глобальна мережа Internet та процеси, що в ній відбуваються. Предметом вивчення дисципліни є веб-технології та принципи вебдизайну, а також методи їх використання при розробці сайтів різноманітного призначення. В результаті вивчення курсу студенти повинні: 1. Знати теоретичні основи, методи організації і створення програм для Web; 2. Знати принципи об’єктно - орієнтованого програмування для створення динамічних web – сайтів, керування доступом змінних сторінок; 3. Вміти створювати практичні програмні елементи веб-сторінок та створювати сценарії їх взаємодії на стороні робочої станції; 4. Створювати динамічні сайти з використанням HTML; 5. Опрацьовувати зображення в HTML та PHP; 6. Створювати та працювати з масивами та їх застосуванням при створенні PHP скриптів; 7. Будувати серверні сценарії на мові PHP; 8. Створювати бази даних для web.

Мета – засвоєння студентами сучасних web-технологій і суміжних галузей знань, вивчення та практичне засвоєння методів і засобів створення web-сайтів. 1.2 Завдання вивчення дисципліни Завдання дисципліни полягає в тому, що студенти повинні: – засвоїти термінологію, прийняту в дисципліні, її основні поняття і визначення; – ознайомитись з базовими концепціями і прийомами створення HTML-сторінок; – вивчити способи візуального оформлення вмісту в гіпертекстовому документі; – розширити уявлення про сучасні web-технології; – вивчити принципи застосування здобутих теоретичних знань для вирішення практичних завдань проектування, створення, оформлення зв'язкових HTML-сторінок; – отримати навички у використанні сучасних технологій верстки та оформлення web-сайтів. У результаті вивчення навчальної дисципліни студент повинен знати: – основні теги HTML 4 та HTML 5; – особливості CSS; – поняття CSS 3; – принципи використання блочної структури документа; – синтаксис та

особливості JavaScript; – основи кросбраузерного та кросплатформного верстання; – поняття JavaScript бібліотеки та CSS фреймворка. вміти: – створювати макет сайту; – виконувати верстання web-сайта; – виконувати оформлення HTML- сторінки згідно з вимогами; – виконувати універсалізацію сайта; – розроблювати засоби динамізації web-сайта; – застосовувати в роботі допоміжні засоби, такі як бібліотеки та фреймворки.

РОЗДІЛ 4. РОЗРОБКА ПЛАТФОРМИ ДЛЯ ПІДТРИМКИ ДИСЦИПЛІНИ «ВЕБ-ТЕХНОЛОГІЇ».

В даному розділі спроектовано веб-платформу, яка може використовуватись для підтримки дисципліни «веб-технології». Та використовуючи фреймворк Django виконано її реалізацію.

4.1. Проектування платформи підтримки дисципліни «Веб-технології»

Опираючись. на представлених в попередніх розділах, особливостях викладання дисципліни в умовах дистанційного навчання, в рамках дипломної роботи спроектовано та реалізовано веб-платформу для викладання матеріалів та перевірки отриманих знань.

Для реалізації платформи обрано фреймворк Django – високорівневий відкритий Python-фреймворк для розробки веб-систем. Його використання дозволяє легко розробити необхідні для платформи модулі та реалізувати їх.

Відповідно до розглянутих моделей та технологій дистанційного навчання, розглянутих в попередніх розділах, запропоновано наступну структуру платформи підтримки дисципліни «Веб-технології» яка повинна забезпечити:

- Реєстрацію користувачів;
- Адміністрування кожного уроку;
- Розбиття уроків по категоріям;
- Розробку запитань для кожного уроку;
- Проведення тестування для перевірки знань;
- Відображення статистики користувача після проходження тестування;

– Ведення загальної рейтингової системи для користувачів;

4.2. Реалізація платформи підтримки дисципліни «Веб-технології»

4.2.1. Модель

Відповідно до поставлених завдань та функцій, які повинна забезпечити платформа, для повної реалізації сформовано наступну реляційну базу даних (БД) (рис. 4.1).

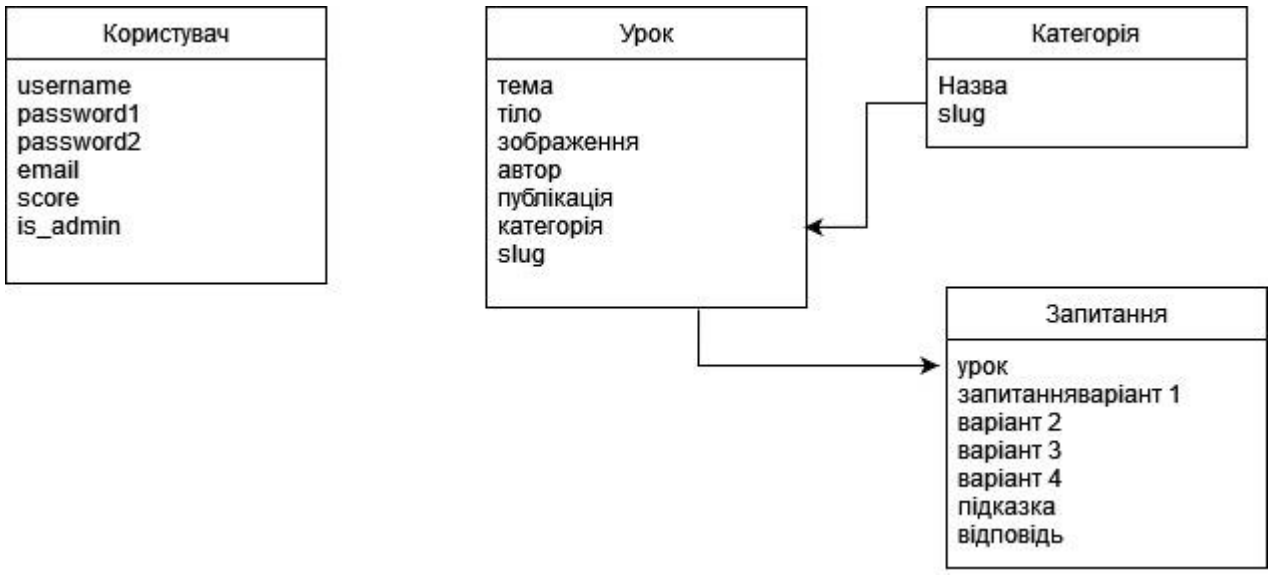

Рисунок 4.1 – Структура бази даних

Відповідно до структурної схеми необхідно реалізувати наступні таблиці:

– Користувач;
– Категорія;
– Урок;
– Запитання;

Використання відкритого фреймворку Django дозволяє використовувати налаштовані внутрішні таблиці для визначення користувачів, проте через необхідність ведення рахунку набраних балів користувачем, її необхідно розширити, добавивши додаткові поля. На рисунку 4.2 зображено програмний код, який відповідає за розширення користувача.

```
class UserProfile(models.Model):
    user = models.OneToOneField(User, on_delete=models.CASCADE)
    avatar = models.ImageField(upload_to='images/users/', verbose_name='Зображення', blank=True)
    score = models.IntegerField(default=0, verbose_name='Рахунок', null=True, blank=True)
    about = models.TextField(verbose_name='Про себе', null=True, blank=True)
    social = models.CharField(max_length=255, verbose_name='Соц мережа', null=True, blank=True)

    def __unicode__(self):
        return self.user

    class Meta:
        verbose_name = 'Профіль'
        verbose_name_plural = 'Профілі'
```

Рисунок 4.2 – Розширення профілю користувача

Таблиця «Категорія» дозволить призводити розбиття уроків на певну категорії, наприклад «HTML», «CSS», «JavaScript» та інші. На рисунку 4.3 зображено програмний код, який відповідає за реалізацію таблиці «Категорія».

```
class Category(models.Model):
    slug = models.SlugField(max_length=255, unique=True, db_index=True, verbose_name='URL')
    name = models.CharField(max_length=255, verbose_name='Назва категорії')

    def __str__(self):
        return self.name

    def get_absolute_url(self):
        return reverse('category', kwargs={'category_slug': self.slug})
```

Рисунок 4.3 – Таблиця «категорія»

Основне значення в платформі відіграє таблиця «Урок», записи в якій призначені для викладення основного матеріалу. На рисунку 4.4 зображено програмний код, який відповідає за реалізацію таблиці «Урок».

```
class Lessons(models.Model):
    category = models.ForeignKey(Category, on_delete=models.CASCADE, verbose_name='Категорія')
    theme = models.CharField(max_length=100, verbose_name='Назва Уроку')
    slug = models.SlugField(max_length=255, unique=True, db_index=True, verbose_name='URL')
    image = models.ImageField(upload_to='lessons/%Y', verbose_name='Зображення', blank=True)
    body = models.TextField(verbose_name='Основа інформація уроку')
    author = models.CharField(max_length=100, verbose_name='Автор')
    is_published = models.BooleanField(default=True, verbose_name='Публікація')

    def __str__(self):
        return self.theme

    def get_absolute_url(self):
        return reverse('lesson', kwargs={'lesson_slug': self.slug})
```

Рисунок 4.4 – «Таблиця Урок»

Для перевірки знань в платформі реалізовано проведення тестування після кожного уроку. Реалізація моделі, яка відповідає за збереження запитань та відповідей на них зображена на рисунку 4.5.

```
class QuesModel(models.Model):
    question = models.CharField(max_length=200, null=True)

    op1 = models.CharField(max_length=200, null=True)
    op2 = models.CharField(max_length=200, null=True)
    op3 = models.CharField(max_length=200, null=True)
    op4 = models.CharField(max_length=200, null=True)

    answer = models.CharField(max_length=200, null=True)
    lesson = models.ForeignKey(Lessons, on_delete=models.CASCADE, verbose_name='Урок')

    hint = models.CharField(max_length=255, verbose_name='Підказка')

    def __str__(self):
        return self.question
```

Рисунок 4.5 – Таблиця «Запитання»

4.2.2. Шаблони

Фреймворк Django дозволяє використовувати шаблонізатор Jinja для забезпечення відсутності дублювання коду та відображення інформації з БД в розроблених HTML шаблонах. На рисунку 4.6 зображено головний шаблон, який буде використовуватись в всіх подальших сторінках.

```
<!doctype html>
<html lang="en">
<head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <!-- Bootstrap CSS -->
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet"
          integrity="sha384-1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3" crossorigin="anonymous">

    <title>{% block title %} WebLesson {% endblock %}</title>
</head>

<body>

{% include 'Lessons/static/main_navbar.html' %}

{%  if messages %}
    {% for message in messages %}
        <div class="alter alter-{{message.tags}}" role="alert">{{message}}</div>
    {% endfor %}
{% endif%}

{% block body %}

<div id="carouselExampleCaptions" class="carousel slide" data-bs-ride="carousel"...>
```

Рисунок 4.6 – Основний шаблон

В ньому реалізовано підключення Bootstrap з хмари, яке в подальшому буде забезпечувати графічне оформлення будь-якої сторінки. Для забезпечення відображень генеруємих повідомлень, в ньому реалізовано механізм їх виведення. Додатково реалізовано механізм пагінації (розбивки на сторінки), та добавлено додаткові JS-скрипти, які будуть відпрацьовувати, при необхідності на будь-якій подальшій сторінці (рис. 4.7).

```
<script>var popoverTriggerList = [].slice.call(document.querySelectorAll('[data-bs-toggle="popover"]'))
var popoverList = popoverTriggerList.map(function (popoverTriggerEl) {
  return new bootstrap.Popover(popoverTriggerEl)
})
</script>
<script>
var alertList = document.querySelectorAll('.alert')
var alerts =  [].slice.call(alertList).map(function (element) {
  return new bootstrap.Alert(element)
})
</script>
<script>
            console.log('hello world')
            const timer=document.getElementById('displaytimer')
            console.log(timer.textContent)
            const inputtag = document.getElementById('timer')

            t=0
            setInterval(()=>{
                t+=1
                timer.innerHTML ="<b>Timer: " +t+" seconds</b>"
                inputtag.value = t
            },1000)
```

Рисунок 4.7 – Додаткові JS-скрипти

Шаблони, які забезпечують відображення форм реєстрації нових користувачів та входу користувачів в особистий профіль зображені на рисунку 4.8 та 4.9.

```
{% extends 'Lessons/index.html' %}

{% block title %} {{block.super}}:: REGISTER {% endblock %}

{% block body %}

<div class="mx-5 mt-5 pt-5">

    <div class="mx-5 mt-5">

        <form method="post">
            {% csrf_token %}
            {{form.as_p}}
            <button type="submit" class="btn btn-success">Зареєструватись</button>

        </form>
    </div>
</div>

{% endblock %}
```

Рисунок 4.8 – Шаблон для реєстрації

```
{% extends 'Lessons/index.html' %}

{% block title %} {{block.super }}:: LOGIN {% endblock %}

{% block name %} Вхід користувача {% endblock %}

{% block body %}
<div class="mx-5 mt-5 pt-5">

    <div class="mx-5 mt-5">
        <form method="post">
            {% csrf_token %}
            {{form.as_p}}
            <button type="submit" class="btn btn-success">Увійти</button>

        </form>
    </div>
</div>

{% endblock %}
```

Рисунок 4.9 – Шаблон для входу в систему

Головною сторінкою для користувача є перелік уроків, які можливо пройти. Шаблон для представлення всіх уроків зображено на рисунку 4.10.

```
{% extends 'Lessons/index.html' %}

{% block title %} {{title}} {% endblock %}

{% block name %} Вхід користувача {% endblock %}

{% block body %}
<div class="row mt-2">
    <div class="col-3">
    {% include 'Lessons/static/category_bar.html' %}
    </div>
    <div class="col-7 mt-3">
        {% for item in page_obj %}
        {% if item.is_published %}
        <div class="card mt-1">
            <b class="card-header">{{ item.theme }}</b>
            <div class="card-body">
                <p class="card-text">{{ item.body | truncatewords:100 }}</p>
                <a href="{% url 'les' item.slug %}" class="btn btn-primary">Перейти до уроку</a>
            </div>
        </div>
        {% endif %}
        {% endfor %}
    </div>
</div>
{% endblock %}
```

Рисунок 4.10 – Шаблон для переліку уроків

Для навігації по платформі реалізовано Navbar. В залежності від того чи авторизований користувач в платформі чи ні навігація буде відрізнятись. На рисунку 4.11 та 4.12 зображено шаблон навігаційної панелі не авторизованого користувача

```
        {% else %}
        <li class="nav-item">
            <a class="nav-link active" aria-current="page" href="{% url 'start_page' %}">HOME</a>
        </li>
        <li class="nav-item">
            <a class="nav-link active" aria-current="page" href="#">ABOUT</a>
        </li>

        {% endif %}
    </ul>
```

Рисунок 4.11 – Панель навігації неавторизованого користувача

```
{% else %}
<li class="nav-item">
    <a class="nav-link" href="{% url 'login' %}">LOGIN</a>
</li>
<li class="nav-item">
    <a class="nav-link" href="{% url 'register' %}">REGISTER</a>
</li>
{% endif %}
```

Рисунок 4.12 – Панель навігації неавторизованого користувача

При авторизації користувачу буде надано можливість переглянути список уроків та створювати сортування по категоріям. Також забезпечено форму пошуку по назвах уроку. (рис. 4.13)

```
<form class="d-flex">
    <ul class="navbar-nav me-auto mb-2 mb-lg-0">
        {% if user.is_authenticated %}
        <li class="nav-item">
            <a class="nav-link" href="{% url 'logout' %}">LOGOUT</a>
        </li>
        <form action="{% url 'lessons' %}">
            <input class="form-control me-2" type="search" placeholder="Search" aria-label="Search"
                   name="search">
            <button class="btn btn-outline-success" type="submit">Search</button>
        </form>

        {% else %}
        <li class="nav-item">
            <a class="nav-link" href="{% url 'login' %}">LOGIN</a>
        </li>
        <li class="nav-item">
            <a class="nav-link" href="{% url 'register' %}">REGISTER</a>
        </li>
        {% endif %}
```

Рисунок 4.13 – Шаблон навігаційної панелі для авторизованого користувача

Для відображення запитань використовується шаблон, код якого зображено на рисунку 4.14.

```
{% extends 'Lessons/index.html' %}

{% block title %} Запитання  {% endblock %}


{% block body %}
{% load static %}
<div class="container ">
<h1>Контрольні запитання</h1>

<div align="right " id="displaytimer"><b>Час: 0 секунд</b></div>

<form method='post' action=''>
    {% csrf_token %}
    {% for q  in questions%}
    <div class="form-group">
      <label for="question">{{q.question}}</label>
    </div>
    <div class="form-check">
        <div class="form-check">
            <input class="form-check-input" type="radio" name="{{q.question}}" id="gridRadios1" value="option1" che
            <label class="form-check-label" for="gridRadios1">
                {{q.op1}}
            </label>
        </div>
        <div class="form-check"...>
```

Рисунок 4.14 – Шаблон запитань

Після проходження користувачем запитань відображення результатів тесту відбувається на окремій сторінці. Код шаблону який забезпечує її відображення зображено на рисунку 4.15.

```
{% extends 'Lessons/index.html' %}

{% block body %}

<div class="container ">

    <div class="card-columns" style="padding: 10px; margin: 20px;">
        <div class="card" align="centre " style="width: 32rem; border:5px black solid">
            <img class="card-img-top" src="https://naked-science.ru/wp-content/uploads/2018/04/field_image_38.jpg"
            <div class="card-body">
                <h5 class="card-title">Набрано балів: {{score}}</h5>

                <p class="card-text">Відсоток правильного: {{percent}}%</p>
                <p class="card-text">Часу затрачено: {{time}} seconds</p>
                <p class="card-text">Правильних відповідей: {{correct}}</p>
                <p class="card-text">Не правильних відповідей: {{wrong}}</p>
                <p class="card-text">Всього запитань : {{total}}</p>
                <h5>Всього найкращого</h5>
            </div>
        </div>
    </div>

</div>

{% endblock %}
```

4.15 – Результати відповідей

Для забезпечення змагального характеру між користувачами, розроблено шаблон, на якому забезпечено вивід кількості набраних балів всіма користувачами (рис. 4.16).

```
{% extends 'Lessons/index.html' %}

{% include 'Lessons/static/main_navbar.html' %}

{% block body %}

{% for user in user_list_rating %}
<div class="card mt-3 mx-5">
    <h5 class="card-header">{{user.user}}</h5>
    <div class="card-body">
        <div class="row">

            <div class="col-4">
                {% if user.avatar %}
                <img src="{{user.avatar.url}}" class="img-thumbnail" width="125px" height="125px" >
                {% else%}
                <img src="https://avatarko.ru/img/kartinka/1/avatarko_anonim.jpg" class="img-thumbnail" width="12
                {% endif%}
            </div>

            <div class="col-8">
                <h5 class="card-title"></h5>
                <p class="card-text">Учасник набрав: {{user.score}} балів. </p>

            </div>
        </div>
```

Рисунок 4.16 – Шаблон статистики користувачів

4.2.3. Представлення

Використовуючи фреймворк Django для взаємодії БД та шаблонами використовуються функції, які називаються представлення. Вони приймають обов'язковий параметр request, який являє собою словник з переданими користувачем параметрами.

Представлення, яке забезпечує реєстрацію користувача зображено на рисунку 4.17. Воно забезпечує зчитування введених в форму даних, та формуванню нового запису в БД.

```
def register(request):
    if request.method == 'POST':
        form = UserRegisterForm(request.POST)
        if form.is_valid():
            form.save()
            messages.success(request, 'Успішно зареєстровано')
            return redirect('login')

    form = UserRegisterForm()

    context = {
        'form': form,
    }

    return render(request, 'Lessons/register.html', context)
```

Рисунок 4.17 – Представлення реєстрації

Для входу користувача в особистий кабінет, для проходження навчання створено представлення, яке зображено на рисунку 4.18. Воно забезпечує зчитування введених в форму логіну та паролю та здійснення ідентифікації та авторизації користувача, при успішному результаті користувач буде авторизований в системі, інакше йому буде надано сповіщення про відсутності такого користувача, або неправильному вводі облікові дані.

```python
def user_login(request):
    if request.method == 'POST':
        form = UserLoginForm(data=request.POST)
        if form.is_valid():
            user = form.get_user()

            login(request, user)
            up = UserProfile.objects.filter(user=user).first()
            if not up:
                up = UserProfile()
                up.user = user
                up.save()

            return redirect('lessons')

        else:
            messages.error(request, 'Користувача не знайдено')
    else:
        form = UserLoginForm()
    return render(request, 'Lessons/login.html', context={'form': form})
```

Рисунок 4.18 – Представлення для входу користувача

Для відображення переліку всіх уроків розроблено представлення, код якого зображено на рисунку 4.19.

```python
def all_lessons(request):
    search_query = request.GET.get('search', '')

    if search_query:
        lessons_items = Lessons.objects.filter(theme__icontains=search_query)
    else:
        lessons_items = Lessons.objects.all()
    category_items = Category.objects.all()

    paginator = Paginator(lessons_items, 10)

    page_numb = request.GET.get('page')
    page_obj = paginator.get_page((page_numb))

    context = {'title': 'Уроки',
               'lessons': lessons_items,
               'categories': category_items,
               'page_obj': page_obj,
               }
    return render(request, 'Lessons/lessons.html', context)
```

Рисунок 4.19 – Представлення всіх уроків

Для забезпечення швидкого сортування виконано представлення, яке приймає додатковий параметр, який являється slug категорії та завдяки цьому відображатиме лише ті уроки, які належать певній категорії (рис. 4.20).

```
def lessons_by_category(request, category_slug):
    search_query = request.GET.get('search', '')
    category_items = Category.objects.all()
    this_cat = Category.objects.filter(slug=category_slug).first()

    if search_query:
        lessons_item_by_category = Lessons.objects.filter(category=this_cat, theme__icontains=search_query)

    else:
        lessons_item_by_category = Lessons.objects.filter(category=this_cat)

    paginator = Paginator(lessons_item_by_category, 10)

    page_number = request.GET.get('page')
    page_obj = paginator.get_page(page_number)

    context = {

        'categories': category_items,
        'page_obj': page_obj,
    }
    return render(request, 'Lessons/by_category.html', context)
```

Рисунок 4.20 – Представлення сортування по категоріям

Основну інформацію уроку можливо переглянути завдяки представленню, яке зображено на рисунку 4.21. Воно виймає всю інформацію про урок та відображає її на відповідному шаблоні.

```
def lesson(request, post_slug):
    lesson_item = get_object_or_404(Lessons, slug=post_slug)
    category_items = Category.objects.all()

    context = {
        'lesson': lesson_item,
        'categories': category_items,

    }
    return render(request, 'Lessons/lesson.html', context)
```

Рисунок 4.21 – Представлення окремого уроку

Після кожного уроку користувачу пропонується пройти тест на перевірку знань прочитаного матеріалу. За дану функцію відповідає представлення зображене на рисунках 4.22 і 4.23. Також воно виконує підрахунок отриманих балів користувача та оновлення загальної кількості набраних балів.

```
def questions(request, lesson_slug):
    if request.method == 'POST':
        this_lesson = get_object_or_404(Lessons, slug=lesson_slug)

        questions = QuesModel.objects.filter(lesson=this_lesson)

        user = UserProfile.objects.filter(user=request.user.pk).first()

        user_score = user.score

        score = 0
        wrong = 0
        correct = 0
        total = 0
        for q in questions:
            total += 1
            if q.answer == request.POST.get(q.question):
                score += 10
                correct += 1
            else:
                wrong += 1
```

Рисунок 4.22 – Представлення запитань

```
        percent = score / (total * 10) * 100

        user.score = user_score + score
        user.save()

        context = {
            'score': score,
            'time': request.POST.get('timer'),
            'correct': correct,
            'wrong': wrong,
            'percent': percent,
            'total': total
        }
        return render(request, 'Lessons/result.html', context)
    else:
        this_lesson = get_object_or_404(Lessons, slug=lesson_slug)

        questions = QuesModel.objects.filter(lesson=this_lesson)

        context = {
            'questions': questions
        }
        return render(request, 'Lessons/questions.html', context)
```

Рисунок 4.23 – Представлення запитань

Представлення загальної статистики відбувається за рахунок представлення, програмний код якого зображено на рисунку 4.24.

```
def leader_board(request):
    user_list_rating = UserProfile.objects.all().order_by('-score')

    context = {
        'title': 'Найкращі учні',
        'user_list_rating': user_list_rating,
    }
    return render(request, 'Lessons/leader_board.html', context)
```

Рисунок 4.24 – Представлення загальної статистики користувачів

Дане представлення відображатиме список користувачів відповідно до кількості набраних ними балів, від найвищого до найнижчого.

4.3. Використання реалізованої платформи підтримки дисципліни «Веб-технології»

4.3.1 Адміністрування веб-платформ

Адміністрування платформи відбувається з допомогою вбудованої в Django панелі адміністратора. Вхід до якої мають тільки користувачі з відповідними правами. На рисунку 4.25 зображено сторінка входу до адміністративної панелі платформи.

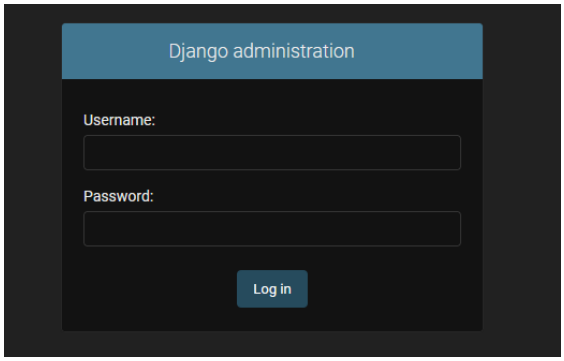

Рисунок 4.25 – Вхід до адміністративної платформи

В адміністративній платформі виконується добавлення контенту, його видалення та редагування. Відповідно до задач платформи можливо створювати нові категорії для нових тем дисципліни, також виконується добавлення нових уроків (рис. 4.26). В полі вводу основного контенту можливо вставляти HTML розмітку, яка буде виконуватись при висвітлені на шаблоні.

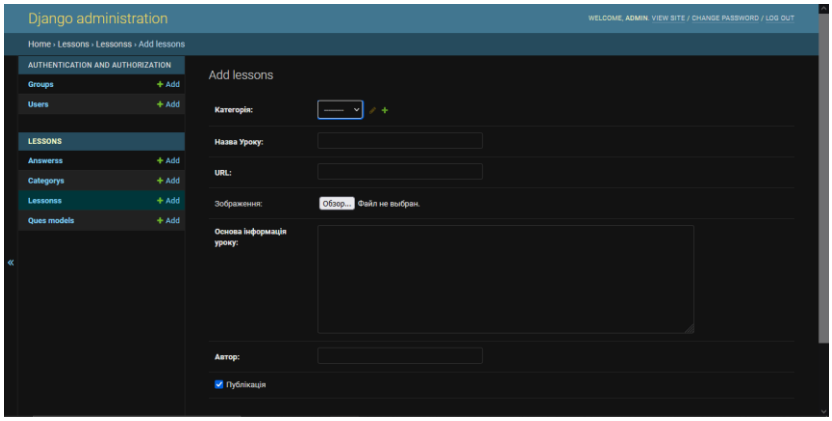

Рисунок 4.26 – Добавлення нового уроку

Для кожного уроку в адміністративній панелі можливо добавляти запитання, які будуть запропоновані по завершенню його проходження (рис. 4.27).

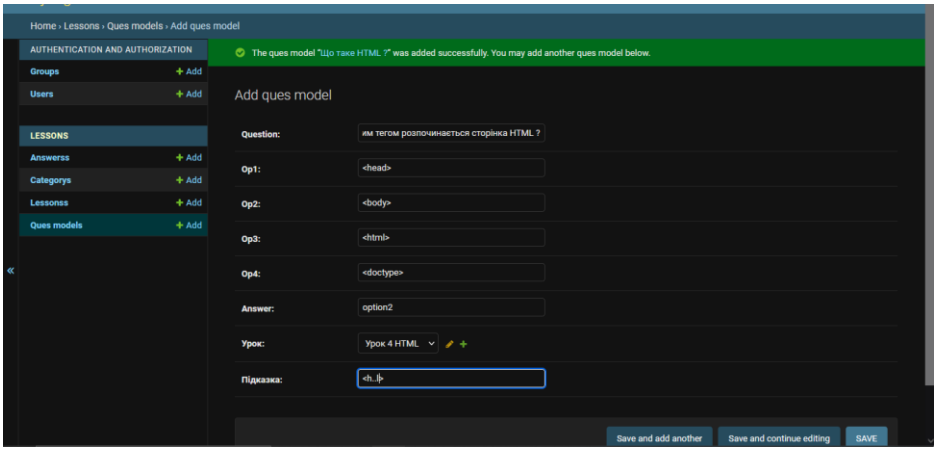

Рисунок 4.27 – Добавлення тестового запитання

4.3.2. Головна сторінка

Стартова сторінка для користувачів несе ознайомлення з платформою та її призначенням. В залежності від того, чи ввійшов користувач в особистий кабінет чи ні сторінка буде змінюватись. На рисунку 4.28 зображено головну сторінку не авторизованого користувача, на ній присутні тільки слайди для ознайомлення та можливість зареєструватись або увійти в особистий кабінет.

Рисунок 4.28 – Головна сторінка не авторизованого користувача

Після успішного входу користувача (рис. 4.29), в нього відкривається можливість перейти на сторінку з переліком уроків, та сторінку перегляду успішності.

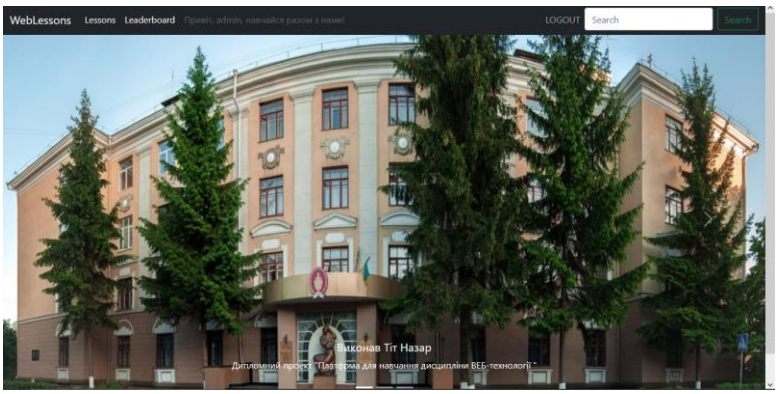

Рисунок 4.29 – Головна сторінка авторизованого користувача

4.3.3.Реєстрація та авторизація користувачів

Реєстрація нових користувачів виконується завдяки вбудованих механізмів Django. Для успішної реєстрації необхідно вказати логін (ім’я користувача) пошту та пароль, який повинен відповідати вимогам безпеки (рис. 4.30).

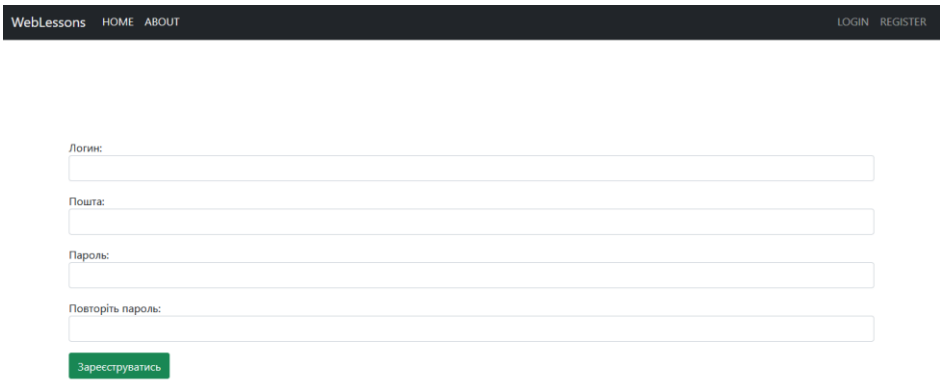

Рисунок 4.30 – Форма реєстрації нового користувача

Для входу в систему, як і для реєстрації використовуються стандартні механізми Django. Форма, для входу в особистий кабінет, зображена на рисунку 4.31.

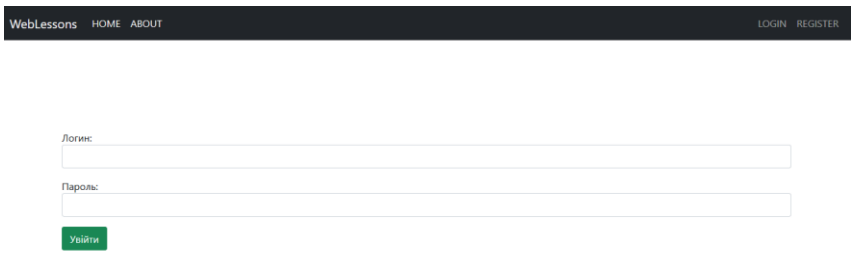

Рисунок 4.31 – Форма входу користувача

4.3.4. Навчання дисципліни «веб-технології» на платформі

Навчання на платформі відбувається за рахунок проходження уроків. На рисунку 4.32 зображено сторінку, на якій відображається список всіх опублікованих уроків.

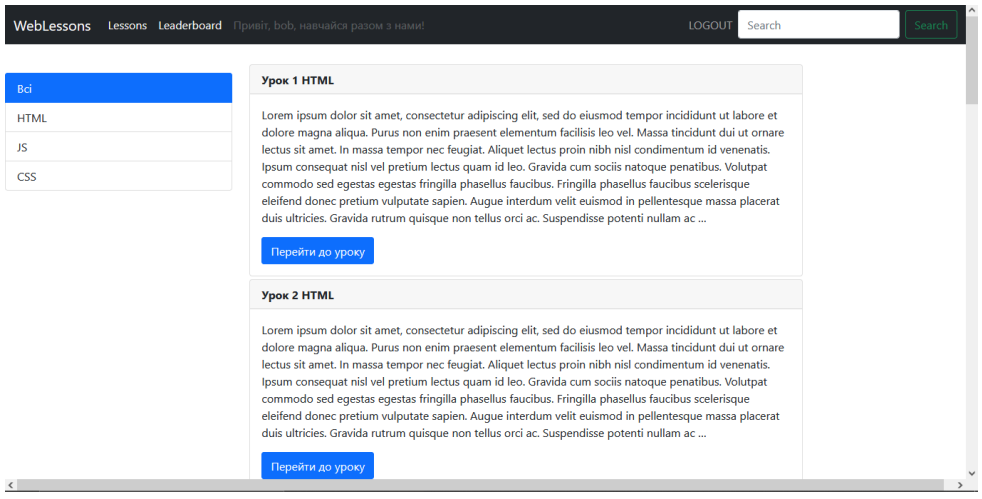

Рисунок 4.32 – Перелік опублікованих уроків

На даній сторінці можливо проводити сортування уроків по категоріям, та проводити пошук по назвам уроку (рис. 4.33).

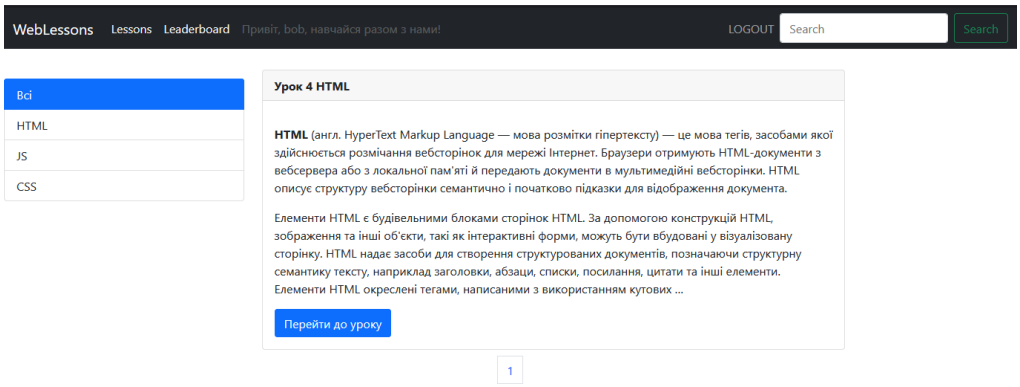

Рисунок 4.33 – Сортування списку уроків

На рисунку 4.34 зображено урок. Кожному уроку можливо задати зображення та тіло уроку. В тілі уроку підтримуються теги HTML, для можливості його кастомізації.

Рисунок 4.34 – Урок в платформі

Після кожного уроку, при наявності запитань, користувачу буде запропоновано перейти за посиланням для їх вирішення (рис. 4.35).

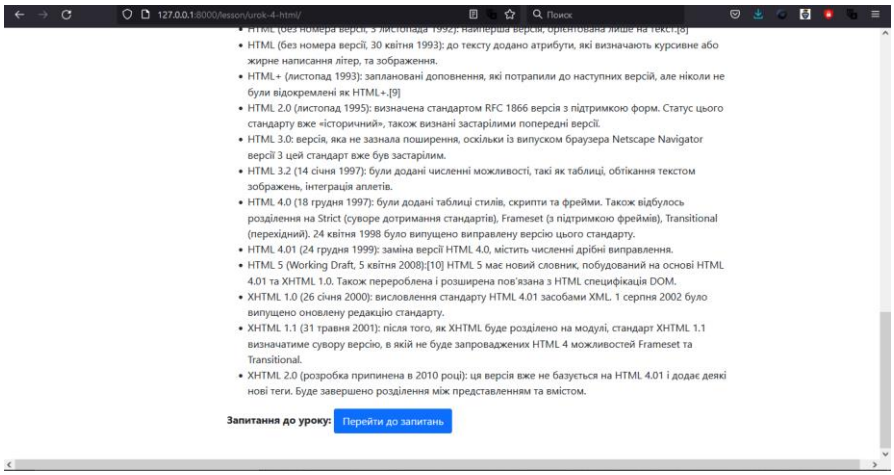

Рисунок 4.35 – Запитання для користувача

4.3.5. Перевірка знань учнів

Для перевірки знань засвоєного матеріалу для користувача запропоновано тестові запитання. Форма для проведення опитування зображено на рисунку 4.36.

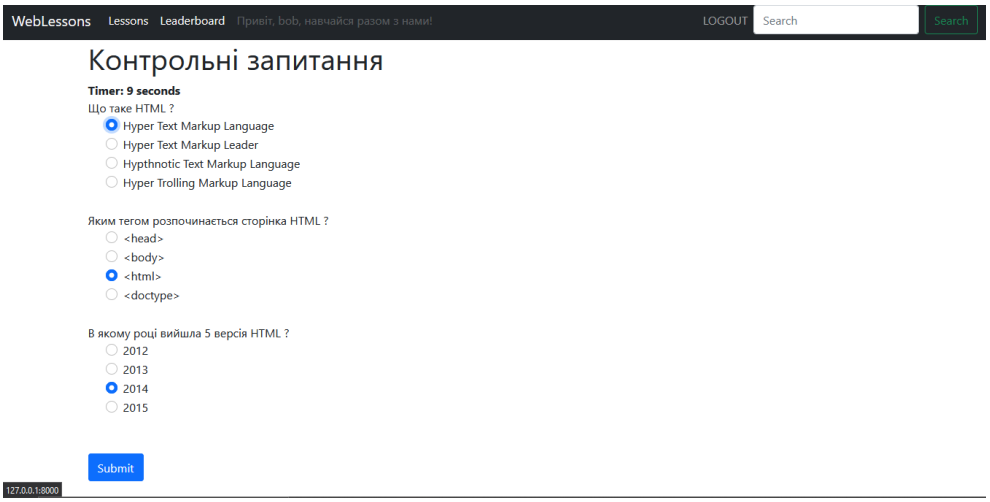

Рисунок 4.36 – Форма для тестування

Надавши відповідь на всі запитання, користувача автоматично перенаправляє на сторінку з результатами (рис. 4.37), на ній показано час, який затратив учень, кількість правильних та не правильних відповідей, набрані бали та відсоток правильних відповідей.

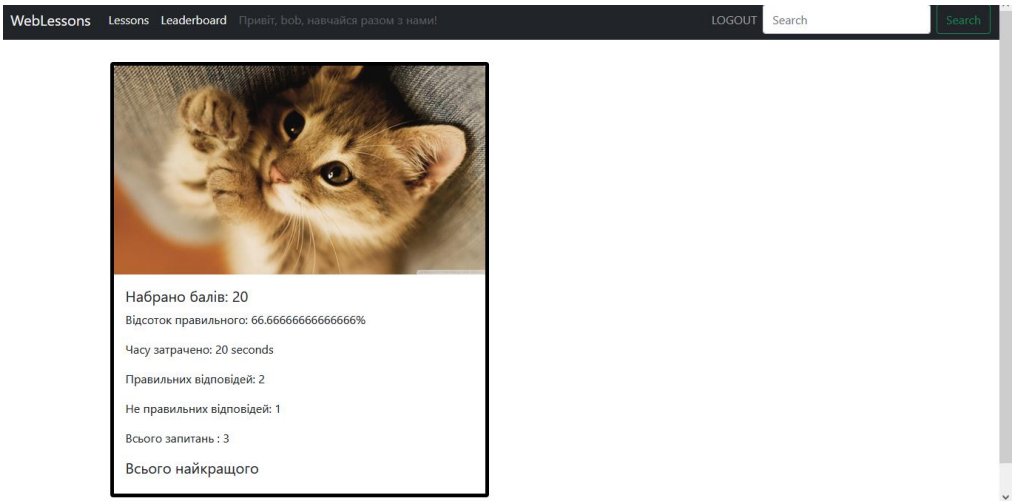

Рисунок 4.37 – Статистика проходження уроку

Для підтримання змагального духу між користувачами, на платформі реалізовано сторінку з загальною кількістю балів користувачами (рис. 4.38). Користувачі з найбільшою кількістю балів знаходитимуться вище в списку.

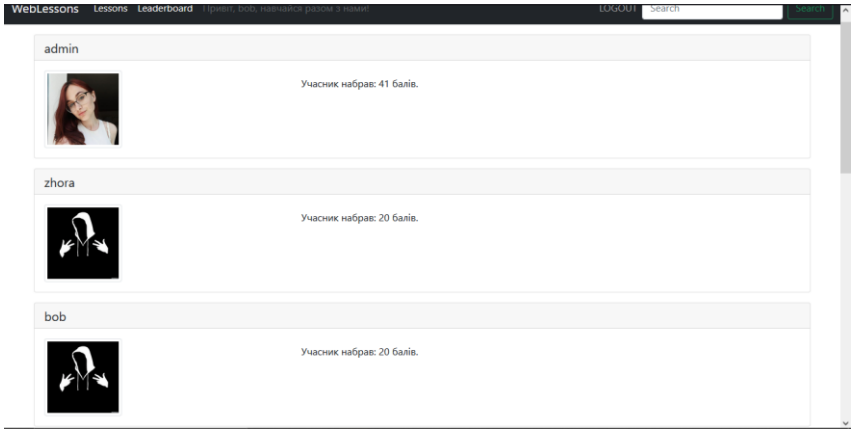

Рисунок 4.38 – Сторінка показу лідерів

ВИСНОВКИ

Дистанційне навчання є перспективною формою організації навчання, в якій студент є суб’єктом навчання. Дистанційна форма навчання спонукає до більшої самостійності студентів та учнів, також підвищує мотивацію навчатися. В роботі описано модель такої форми навчання, учасників, складники, які сприяють організації даної форми навчання. В зв'язку з впровадженням дистанційної форми навчання, намагаються використовувати широкий спектр технічних засобів, комп’ютерну техніку та цифрові електронні засоби комунікації. На такій основі створюються інформаційні системи. Метою ІС є забезпечення інформаційних потреб користувачів.

В даній роботі, спроектовано та реалізовано веб-платформу, яка може використовуватись для підтримки дисципліни «веб-технології». Розглянуто її використання користувачем для вивчення певної дисципліни. Створена платформа розроблена за допомогою використання фреймворку Django. Платформа містить у собі лекційні матеріали, тести для самоконтролю. Також впроваджена система адміністрування, що є перевагою над іншими подібними платформами. Така функціональність дозволяє адміністратору легко додавати та видаляти матеріали, які розміщені в ІС, також це зменшує час створення тестів до певної теми. Інформаційна система має функцію реєстрування користувача. Ще однією особливістю платформи є таблиця кращих користувачів за результатами тестів. Також є можливість пошуку лекції за назвою.

Мета дипломної роботи була досягнута, поставлені завдання були виконані в повному обсязі.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Історичні етапи розвитку інформаційних систем. URL: https://studfile.net/preview/7516525/page:40/ (дата звернення: 15.11.2021)
2. Биков В. Ю. Моделі організаційних систем відкритої освіти : монографія. Київ : Атіка, 2008. 684 с.
3. Гнезділова К. М, Касярум С. О. Моделі та моделювання у професійній діяльності викладача вищої школи : навчальний посібник. Черкаси : Видавець Чабаненко Ю. А., 2011. 124 с.
4. Горбатюк Р. М., Романишена Л. Експериментальна модель дистанційного навчання майбутніх фахівців у вищому навчальному закладі. Наукові записки. Серія: педагогіка. 2016. № 2. С. 68–75.
5. Горбатюк Р. М. Структурно-функціональна модель технології формування педагогічної фасилітації майбутніх інженерів-педагогів. Сучасні інформаційні технології та інноваційні методики навчання у підготовці фахівців: методологія, теорія, досвід, проблеми : збірник наукових праць. Вип. 44. Київ; Вінниця: ТОВ фірма «Планер», 2016. С. 290–294.
6. Кухаренко В. М. Про систему дистанційного навчання у відкритому дистанційному курсі. Інформаційні технології в освіті. 2012. № 11. С. 32–42.
7. Кухаренко В. М. Система дистанційного навчання університету. Теорія та методика навчання математики, фізики, інформатики. Т. XIII (2015), вип. 3 (37) С.220–233.
8. Полат Е. С. Дистанционное обучение. URL: https://gigabaza.ru/doc/101024.html. (дата звернення: 15.11.2021)

9. Полат Е. С. Понятийный апарат дистанционного обучения. URL: http://vio.uchim.info/Vio_19/cd_site/articles/art_1_21.htm. (дата звернення: 15.11.2021)
10. Тверезовський В. А., Лукова-Чуйко Н. В. Інформаційні процеси у вищій школі та дистанційна освіта: на роздоріжжі. Теорія методики навчання математики, фізики, інформатики. Т. XIII (2015), вип. 3 (37). С. 215–219.
11. Технологія створення дистанційного курсу : навчальний посібник. / Биков В. Ю. та ін. ; за ред. В. Ю. Бикова та В. М. Кухаренка. Київ : Міленіум, 2008. 324 с.