

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЇВО-МОГИЛЯНСЬКА АКАДЕМІЯ»

Кафедра інформатики факультету інформатики



ЕЛЕКТРОННИЙ КУРС «ДИТЯЧЕ ПРОГРАМУВАННЯ»

Дипломна робота

за спеціальністю «Інженерія програмного забезпечення» 121

Керівник роботи
доктор фізико-математичних наук,
професор
Глибовець Микола Миколайович

Виконала студентка
Мітіньова А.С.

Індивідуальне завдання до курсової роботи

Міністерство освіти і науки України

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»

Кафедра інформатики факультету інформатики

ЗАТВЕРДЖУЮ

Зав. кафедри
інформатики,

кандидат технічних наук,
доцент

Горохівський С. С.

(підпис)

”_____”
2020 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

на дипломну роботу

студентці Мітіньовій Анастасії Сергіївні факультету інформатики 2 курсу
магістратури

ТЕМА: Електронний курс «Дитяче програмування»

Вихідні дані:

- Дослідження засобів для створення електронних курсів
- Електронний курс створений за допомогою обраного засобу

Зміст ТЧ до курсової роботи:

Індивідуальне завдання

План виконання курсової роботи

Анотація

Вступ

1. Електронне, дистанційне навчання

2. Електронний курс

3. Засоби розробки електронних курсів

4. Електронний курс «Дитяче програмування»

5. Програмна реалізація електронного курсу

Висновки

Список використаних джерел

Дата видачі „____” _____ 2020 р. Керівник _____
(підпис)

Завдання отримав _____

План виконання курсової роботи

Тема: Електронний курс «Дитяче програмування»

Календарний план виконання роботи:

| № п/п | Назва етапу дипломного проекту (роботи) | Термін виконання етапу | Примітка |
|----------|---|------------------------------|----------|
| 1 | Отримання завдання на курсову роботу. | 04.10.2020 | |
| 2 | Знайомство з теоретичними основами, збір інформації | 08.11.2020 | |
| 3 | Поглиблене вивчення літератури за темою | 09.01.2021 | |
| 4 | Підготовка тестів для модульного контролю учнів | 12.03.2021 | |
| 5 | Створення прототипу вебсайту, для розміщення електронного курсу | 12.04.2021 | |
| 6 | Створення слайдів та аналіз отриманих результатів з керівником | 10.05.2021 | |
| 7 | Попередній захист | 14.05.2021 | |
| 8 | Корегування роботи за результатами попереднього захисту. | 20.05.2021 | |
| 9 | Остаточне оформлення пояснювальної роботи та слайдів. | 03.05.2021 | |
| 10 | Захист дипломної роботи | 15.06.2021 | |

Студент _____

Керівник _____

“ ”

Зміст

| | |
|---|----|
| АНОТАЦІЯ | 7 |
| ВСТУП | 8 |
| 1 ЕЛЕКТРОННЕ, ДИСТАНЦІЙНЕ НАВЧАННЯ | 10 |
| 1.1 Історія розвитку електронного навчання | 10 |
| 1.2 Що таке електронне та дистанційне навчання? | 14 |
| 1.3 Сучасні тенденції | 16 |
| 1.3. Топ-3 рішень для електронного навчання | 19 |
| 2 ЕЛЕКТРОННИЙ КУРС | 22 |
| 2.1 Доступ до електронного курсу | 22 |
| 2.3 Структура модуля електронного курсу | 24 |
| 3 ЗАСОБИ РОЗРОБКИ ЕЛЕКТРОННИХ КУРСІВ | 26 |
| 3.1 Стандартні засоби | 26 |
| 3.1.1 Moodle | 26 |
| 3.1.2 Moodle Cloud | 27 |
| 3.1.3 iSpring Suite | 27 |
| 3.1.4 iSpring Learn | 27 |
| 3.2 Google Forms | 28 |
| 4 ЕЛЕКТРОННИЙ КУРС «ДИТЯЧЕ ПРОГРАМУВАННЯ» | 34 |
| 4.1 Призначення курсу | 34 |
| 4.2 Зміст електронного курсу | 34 |
| 4.2.1 Загальні відомості | 34 |

| | |
|--|----|
| 4.2.2 Тести для модульного контролю | 36 |
| 4.2.3 Практичні завдання для модульного контролю | 40 |
| 4.2.4 Теоретичний навчальний матеріал | 40 |
| 4.3 Сторінка викладача | 52 |
| 4.4 Інструкція для вивчення електронного курсу | 53 |
| 5 ПРОГРАМНА РЕАЛІЗАЦІЯ ЕЛЕКТРОННОГО КУРСУ | 55 |
| 5.1 Обраний інструментарій | 55 |
| 5.1.1. Мова Java | 55 |
| 5.1.1.1 Загальний огляд | 55 |
| 5.1.1.2 Встановлення на комп'ютер | 55 |
| 5.1.1.3 Обрана версія Java | 55 |
| 5.1.2 Spring Boot | 55 |
| 5.1.2.1 Огляд Spring Boot | 55 |
| 5.1.2.2 Переваги Spring Boot | 56 |
| 5.1.2.3 Як працює Spring Boot? | 57 |
| 5.1.3 IntelliJ IDEA | 58 |
| 5.1.3.1 Загальний огляд | 58 |
| 5.1.3.2 Встановлення на комп'ютер | 58 |
| 5.1.4 Thymeleaf | 58 |
| 5.1.5 HTML | 58 |
| 5.1.6 Bootstrap | 59 |
| 5.1.6.1 Загальний огляд | 59 |

| | |
|--|----|
| 5.1.6.2 Підключення | 60 |
| 5.1.6.3 З чого складається Bootstrap? | 60 |
| 5.1.6.4 Обрана версія Bootstrap | 60 |
| 5.1.7 Apache Maven | 60 |
| 5.1.8 Apache Tomcat | 61 |
| 5.1.9 jQuery | 61 |
| 5.1.10 JavaScript | 61 |
| 5.2 Особливості програмної реалізації | 62 |
| 5.2.1 Створення початкового коду для веб-сайту | 62 |
| 5.2.1 Модель-Вигляд-Контроллер | 62 |
| 5.3 Основні компоненти | 62 |
| 5.4 Інструкція з локального розгортання проекту з веб-сайтом | 64 |
| ВИСНОВКИ | 66 |
| Використані джерела | 67 |
| Додаток А | 68 |

АНОТАЦІЯ

Мітіньова Анастасія

Електронний курс «Дитяче програмування»

Інженерія програмного забезпечення

Національний університет «Києво-Могилянська академія»

Київ, 2021 рік

Мета дипломної роботи – створити основу для електронного курсу «Дитяче програмування». Дипломна робота містить такі розділи:

1. Електронне, дистанційне навчання
2. Електронний курс
3. Засоби розробки електронних курсів
4. Електронний курс «Дитяче програмування»
5. Програмна реалізація електронного курсу

У процесі виконання роботи досліджено теоретичний матеріал, який стосується електронного, дистанційного навчання, електронного курсу, створено основу для електронного курсу «Дитяче програмування»

Робота містить 4 таблиці, 17 малюнків та 16 використаних джерел.

Електронне навчання, дистанційне навчання, електронний курс, дитяче програмування.

ВСТУП

Технологічні досягнення значною мірою вплинули на різні галузі й сектор освіти не став виключенням. Диджиталізація здійснює вплив на традиційні підходи навчання, і, як наслідок, індустрія освіти продовжує розвиватися, що робить її легкодоступною для кожного та надає можливість студентам навчатися у будь-який час та в будь-якому місці.

Сьогодні навчання вийшло за рамки класної кімнати. Впровадження електронного навчання і систем управління навчанням стають все більш популярними, оскільки все більше навчальних закладів досліджують сучасні підходи до викладання та навчання. Як результат вплив електронної освіти та систем управління навчанням зростає і, як очікується, буде зростати й надалі в найближчі роки.

Через пандемію коронавірусу навчальні заклади опановують інструменти електронного навчання, щоб забезпечити неперервну освіту студентам дистанційно через Інтернет. Це стосується і світових університетів, які впроваджують онлайн-навчання та пропонують великий вибір онлайн-курсів, які допомагають студентам вдосконалити свої знання з обраних дисциплін. Застосування інструментів навчання в освітньому секторі сприяє зростанню ринку навчання. Аналогічним чином, технології систем управління навчанням набувають все більшої популярності на ринку освітніх послуг, оскільки попит на дистанційне навчання та якісну освіту зростає у глобальному масштабі. Завдяки платформам, які спеціалізуються на управлінні навчанням та забезпечують доставлення, відстеження та управління електронним вмістом, не дивно, що сьогодні на ринку спостерігається позитивне зростання. Спостерігаються наступні тенденції:

- У 2016 році ринок електронного навчання було оцінено в 46 мільярдів доларів і, як очікується, він зросте до 243 мільярдів доларів до 2022 року [1].
- У 2019 році ринок електронного навчання вже перевищив 200 мільярдів доларів [1].

- Ринок мобільного навчання у 2020 році було оцінено у 38 мільярдів доларів [1].
- Очікується, що до 2021 року ринок електронного навчання зросте до 33.5 мільярдів доларів, а до 2026 року – перевищить 375 мільярдів доларів [1].
- До 2023 року світовий ринок систем управління навчанням оцінюватиметься у 22,4 мільярда доларів [1].

Тож з описаного вище можна зробити висновки, що розробка електронних курсів є актуальною та важливою темою, тому метою даної дипломної роботи було дослідження програмного інструментарію створення електронних курсів (електронний курс на базі книги М. М. «Вступ до дитячого програмування. Мова JavaScript»), розробка електронного курсу, яка включає тестування та розробка прототипу сайту курсу.

Об'єктом дослідження у цій роботі був процес створення електронного курсу, а предметом дослідження були програмні інструментальні засоби створення та підтримки електронних курсів.

Наукова значимість результатів отриманих в процесі роботи над дипломною роботою в тому, що було запропоновано методологію створення електронного курсу на базі Google Forms та Spring Boot, а практична значимість результатів – полягає у можливості їхнього використання у навчальному курсі для дітей та дорослих, які бажають розбиратися у будові та роботі комп'ютера, опанувати базові основи програмування, веб-програмування, мову програмування JavaScript, технології HTML, CSS, платформу Node.js, технології Ajax, JSON та бібліотеку jQuery.

Текстова частина даної роботи має п'ять розділів. У першому розділі визначається електронне та дистанційне навчання. У другому розділі описується визначення електронного курсу, його структура. У третьому розділі описуються засоби створення електронних курсів. Четвертий розділ присвячено опису створеного електронного курсу (тестування для модульного контролю +

теоретичний матеріал). А п'ятий розділ присвячено опису програмної реалізації електронного курсу.

1 ЕЛЕКТРОННЕ, ДИСТАНЦІЙНЕ НАВЧАННЯ

1.1 Історія розвитку електронного навчання

Зародження е-навчання почалося майже 200 років назад, але до моменту розповсюдження електронно-обчислювальних машин розвивалося слабо. З середини 20-ого століття електронне навчання почали активно застосовувати в університетах, а в 90-і роки воно вийшло за їх межі і стало невід'ємною частиною освіти і підвищення кваліфікації у всіх сферах людської діяльності. В першу чергу, це було пов'язано з активним розвитком цифрових технологій. Хронологія подій розміщена в таблиці 1.1.

Таблиця 1.1

| Рік | Подія |
|------|--|
| 1728 | Перший документально зафіксований приклад масового дистанційного навчання (d-learning) – американський професор Калеб Філіпс (англ. Caleb Phillips) організував заочні курси стенографії через газету Boston Gazette [2]. |
| 1840 | Дистанційне навчання стало двостороннім – відомий британський лінгвіст Айзек Пітман (англ. Isaac Pitman), у той час шкільний вчитель в Бостоні, почав не тільки розсилати студентам завдання (також з області стенографії), але і отримувати їх назад для перевірки [2]. |

| Рік | Подія |
|------|--|
| 1856 | Метод вийшов за межі навчання стенографії – німець Густав Лангенштейдт (нім. Gustav Langenscheidt) та француз Шарль Туссен (фр. Charles Toussaint) створили першу школу дистанційного навчання іноземним мовам, яка виявилася достатньо успішною [2]. |
| 1909 | У повісті-антиутопії «Машина зупиняється» англійський письменник Едвард Морган Форстер (англ. Edward Morgan Forster) передбачив появу Інтернету і, зокрема, його використання для проведення лекцій та екзаменів на відстані [2]. |
| 1920 | Сідней Пріссі (англ. Sidney Pressy), викладач університету Огайо, створив першу машину для тестування ємністю в 100 питань. Цей винахід дозволяв студентам перевіряти засвоєння знань самостійно, а також «змушувало» їх думати над помилками, а не уникати їх (тест можна було продовжити тільки після вводу правильної відповіді) [2]. |

| Рік | Подія |
|------|--|
| 1960 | <p>Дональдом Бітцером (англ. Donald Bitzer) з Іллінойського університету створена перша справжня електронна система навчання на основі комп'ютера. Вона отримала назву PLATO і призначалася для підвищення грамотності студентів американських вищих навчальних закладів. Система мала значимі для свого часу функції, такі як графічний термінал з високим розширенням, можливість доступу до електронних навчальних ресурсів, навчальні ігри, і навіть дозволяла викладачам і студентам передавати один одному повідомлення за допомогою чату та форуму [2].</p> |
| 1960 | <p>Патрік Суппес (англ. Patrick Suppes) і Річард С. Аткинсон (англ. Richard C. Atkinson) з Стенфордського університету провели експерименти по навчанню учнів молодшої школи читанню і обрахунку за допомогою перших комп'ютерів [2].</p> |
| 1969 | <p>У Великобританії відкрився перший Відкритий університет – вищий учбовий заклад, повністю направлене на заочне навчання [2].</p> |

| Рік | Подія |
|-------------|--|
| 1971 | Австрійський філософ і соціальний критик Іван Ілліч (нім. Ivan Illich) опублікував знакову книгу «Звільнення від шкіл. Пропорційність і сучасний світ» (Deschooling Society), де описав Інтернет, який «навчає», як модель, яка дозволяє студенту вибудовувати освітній процес для себе [2]. |
| 1974 - 1978 | Спеціалісти Технологічного інституту Нью-Джерсі Мюррей Турроф (англ. Murray Turoff) і Старр Роксання Хілц (англ. Starr Roxanne Hiltz) розробив електронну систему обміну інформацією (EIES – Electronic Information Exchange System), яка дозволила проводити сеанси конференц-зв'язку [2]. |
| 1986 | Лінда Харасім (англ. Linda Marie Harasim), професор Школи комунікації при Університеті Саймона Фрезера, провела перший в світі онлайн-курс разом з канадським соціологом Дороті Едіт Сміт (англ. Dorothy Edith Smith) на тему комп'ютерної освіти жінок [2]. |
| 1994 | Освітня організація CompuHigh заснувала першу в світі онлайн-навчання Whitmore School [2]. |

| Рік | Подія |
|------|--|
| 1997 | Майкл Часен (англ. Michael ChaSen) і Меттью Пітінські (Matthew Pittinsky) розробили систему управління навчанням Blackboard Learning Management System, основна ціль якої адміністрування онлайн-курсів [2]. |
| 1999 | На семінарі з комп'ютерного навчання (CBT Systems) у Лос-Анджелесі був вперше використаний і введений в побут професійної середі термін електронне навчання – e-learning [2]. |

1.2 Що таке електронне та дистанційне навчання?

Не дивлячись на багату історію розвитку, у світі до тепер не існує єдиного визначення для поняття електронне навчання, яке було б прийнято багатьма спеціалістами. Існує плутаниця через введення в 1995 році таких термінів, як «дистанційна освіта» і «дистанційні освітні технології».

У результаті сьогодні існує багато людей, які вважають, що електронна та дистанційна освіта – це аналогічні поняття, але насправді це не так, тому для запобігання непорозумінь варто дати визначення цим поняттям.

Найчастіше, під час пояснення терміну електронне навчання використовують визначення, яке надали спеціалісти ЮНЕСКО: «навчання за допомогою Інтернет і мультимедіа», але існують і інші, які найбільш наочно відображають основні принципи електронного навчання [2].

Наприклад, Марк Дж. Розенберг (англ. Marc J. Rosenberg), доктор філософії дав в 2001 році наступне визначення: електронне навчання – «використання інтернет-технологій для надання широкого спектра рішень, які забезпечують підвищення знань і продуктивності праці» [2].

Міхаель Керрес (нім. Michael Kerres), професор німецького університету Дуйсбург-Ессен, визначив поняття «електронне навчання», як «усі види навчання, у процесі якого приймають участь електронні або цифрові пристрої, які представляють і розподіляють освітні матеріали і підтримують міжлюдські комунікації» [2].

Професор Є. С. Полат визначає дистанційну освіту, як самостійну форму навчання, за якої взаємодія вчителя та учнів, а також учнів між собою здійснюється на відстані і відображає всі властиві навчальному процесу компоненти (цілі, вміст, методи, організаційні форми, засоби навчання), які реалізуються специфічними засобами інтернет-технологій або інших інтерактивних технологій [2].

Професор А. А. Андреев формулює дистанційне навчання, як «цілеспрямований, організований процес інтерактивної взаємодій учнів між собою із засобами навчання, інваріантний до їх розташування в просторі і часі, який реалізується в специфічній дидактичній системі» [2].

Дистанційна освіта визначається ним як «система, у якій реалізується процес дистанційного навчання і визначається індивідуумом досягнення і підтвердження освітнього цензу» [2].

У цілому дистанційне навчання та електронне навчання мають загальні точки перетину, але все ж це не синоніми. Дистанційна освіта передбачає, перш за все, те, що вчитель та учні знаходяться на відстані, при цьому цілковито неважливо, у якому вигляді (писемний і електронний) використовуються і яким чином доставляються навчальні матеріали (пошта, кур'єр, інтернет-технології, телебачення). У свою чергу, електронне навчання не обов'язково здійснюється на відстані, але передбачає використання навчальних матеріалів тільки в електронному вигляді, що реалізується за допомогою цифрових технологій. Електронне навчання може здійснюватися за допомогою персонального комп'ютера, смартфона, VR-тренажера і іншими способами.

1.3 Сучасні тенденції

Останнім часом стали популярними і з кожним роком знаходять все більш широке розповсюдження наступні тенденції в електронному навчанні наведено в таблиці 1.2.

Таблиця 1.2

| Назва тенденції | Суть тенденції |
|--|---|
| Адаптивне або персоналізоване навчання | Модель, основною ідеєю якою є підстроювання під здібності, знання, уміння і навіть настрій учня. |
| Мікронавчання | Ґрунтується на наступних принципах: тривалість кожної одиниці контенту від 1 до 15 хвилин, фокус на результаті навчання, можливістю транслявати контент на будь-яких платформах і пристроях. При такому навчанні мозок учня не перевантажується, і навчальний матеріал засвоюється краще. |

| Назва тенденції | Суть тенденції |
|--|--|
| Розвиток гнучких навичок (від англ. soft skills) | Таких як лідерство, креативність, відповідальність, критичне мислення, дисциплінованість, робота в команді і інше. Щорічно все більше задач автоматизують, тому за оцінками Всесвітнього економічного форуму вже до 2022 року 50% усіх працівників зажадають перепідготовки. Професійні навички (від англ. hard skills – жорсткі навички) уходять на другий план, і в майбутньому під час працевлаштування ключову роль будуть мати саме гнучкі навички. |
| Гейміфікація | Використання в освітніх цілях ігрових підходів, які широко розповсюджені в комп'ютерних іграх. Отримання балів і «ачівок» (від англ. to achieve – досягати, добиватися чогось) стимулює до особистих досягненням. |

| Назва тенденції | Суть тенденції |
|--|---|
| Віртуальна реальність (VR від англ. virtual reality – штучна реальність) | Створений технічними засобами світ, що взаємодіє з людиною за допомогою таких відчуттів, як зір, слух, дотик і інші. Основною перевагою VR є вироблений ефект занурення, який полягає в тому, що учень перестає відчувати себе зовнішнім спостерігачем і починає сприймати віртуальний світ, як справжній. |
| Доповнена реальність (AR від англ. augmented reality) | це технологія додавання до відчуттів з реального світу уявних об'єктів з метою доповнення одержаних відомостей і поліпшення сприйняття інформації. Простіше кажучи, на зображення реального світу накладається додаткова інформація. Використання систем доповненої реальності в освітній діяльності дозволяє відтворювати неймовірні процеси і явища, які фізично неможливо реалізувати в реальному житті. |

| Назва тенденції | Суть тенденції |
|------------------|---|
| Штучний інтелект | властивість штучно-створених систем і програм виконувати творчі функції, які традиційно вважаються прерогативою людини. Найбільш поширеним варіантом використання штучного інтелекту є чат-боти (від англ. Chatbot) - програми, які з'ясовують потреби користувачів, а потім допомагають задовольнити їх. |

Яскравим прикладом чат-бота є Siri – розумний помічник для володарів пристроїв Apple. Джо Ганчі (від англ. Joe Ganci), президент американської консалтингової і навчальної компанії електронного навчання eLearningJoe, підкреслює: «Чат-боти здатні принести величезну користь для навчання. Наприклад, можна посилити діалоговий тренажер штучним інтелектом. Уявіть, наскільки складнішою і реалістичнішою стане задача з відпрацювання заперечень віртуальних клієнтів» [2].

Також останнім часом все більше застосування знаходять технології e-learning 2.0 (електронне навчання, основане на технології Web 2.0, коли контент створюють самі користувачі за допомогою блогерів, wiki, подкастів і соціальних мереж) і m-learning (навчання за допомогою мобільних пристроїв).

1.3. Топ-3 рішень для електронного навчання

Для кращого розуміння визначення електронного навчання варто розглянути рішення, які сьогодні найбільш часто використовують учбові заклади під час підготовки школярів та студентів, а також компанії під час навчання і підготовки спеціалістів.

Перше місце в цьому рейтингу, безумовно, займають електронні курси, які являють собою тематично завершений, структурований учбовий матеріал,

що включає цілі навчання, відповідний їм освітній контент і інструменти підсумкового контролю. Популярність курсів обумовлена їх невеликою вартістю, простотою розробки і високою ефективністю навчання. Для більшості з них не потребується навіть наявність гарного комп'ютеру або Інтернету. Навчання на базі електронних курсів може проводитися в форматі як online, так і offline.

На другому місці за популярністю знаходяться відеоконференції і відеоуроки, які представляють собою вид групових телекомунікацій між групою людей. У випадку відеоконференцій усі учасники бачать і чують один одного, тоді як під час відео уроку бачать і чують лише одного лектора, а він, при цьому бачить і чує усіх, хто навчається. Заняття на базі відеотрансляцій сьогодні використовуються усюди. Завдяки ним можна навчитися готувати, робити манікюр, вивчити іноземні мови, опанувати програмування, позайматися спортом з інструктором та багато іншого.

Третє місце в рейтингу рішень для електронного навчання займають тренажери і симулятори. Саме завдяки ним студенти і спеціалісти по всьому світу можуть відпрацьовувати практичні навички. Важливим аспектом є те, що при навчанні на тренажерах і симуляторах учні практично не відчують стрес, так як не бояться зламати дороге устаткування при прокатці металу, підірвати завод під час перекачування нафтопродуктів або вбити людей при керуванні літаком.

Найвища ефективність і незамінність тренажерів при підготовці деяких фахівців давно вже оцінена навчальними закладами та компаніями усього світу. Існує безліч прикладів їх успішного використання.

Наприклад, у компанії Hyundai Mobis після впровадження програми з управління клієнтами (CRM) багато новеньких через складність інтерфейсу почали здійснювати помилки під час обробки замовлень. Збитки компанії доходили до декількох мільйонів доларів за неправильну дію. Для вирішення цієї проблеми компанія впровадила симулятор по роботі з CRM [2].

У вересні 2018 року американська компанія Walmart, яка є найбільшою в світі мережею оптової та роздрібної торгівлі, купила 17000 VR-гарнітур Oculus

Go. Тепер більше одного мільйона її співробітників майже в 5000 магазинах по всьому світу використовують окуляри віртуальної реальності для навчання. Зокрема, їх вчать швидко обслуговувати клієнтів під час «Чорної п'ятниці», коли число клієнтів зашкалює [2].

В корпорації Boeing, яка є найбільшим світовим виробником авіаційної, космічної і військової техніки, інженери за допомогою AR-програми встановлюють електропроводку в літаках. Як відмічають в компанії, це на 25% скорочує час роботи і страхує від помилок [2].

У зв'язку з розвитком торгівлі і сфери послуг багато компаній сьогодні застосовують діалогові тренажери для навчання менеджерів відділу продажів правильним варіантам ведення бесід з клієнтами.

Не дивлячись на великі плюси і успішні приклади, тренажери і симулятори розповсюджені далеко не усюди. Причиною є їх дорожнеча. Не багато організацій мають достатні ресурси для покупки таких навчальних продуктів.

2 ЕЛЕКТРОННИЙ КУРС

Електронний навчальний курс (ЕНК) – це комплекс електронних навчально-методичних матеріалів, створених для організації індивідуального та групового навчання з використанням дистанційних технологій (ДТ), що базуються на Internet-технологіях, відповідно до графіку навчального процесу навчального закладу [3].

Електронний навчальний курс передбачає взаємодію викладача та учнів, які опановують матеріал під керівництвом викладача.

2.1 Доступ до електронного курсу

Електронні навчальні курси розміщуються на навчальному порталі. Як правило, робота порталу організована за допомогою електронної системи MOODLE. Ця система дозволяє учням, за допомогою глобальної мережі Інтернет, ознайомлюватись з навчальними матеріалами, які створюються у вигляді різнотипних інформаційних ресурсів, таких як: текст, відео, анімація, презентація, електронний посібник тощо, а також проходити тестування, виконувати завдання та відправляти їх на перевірку викладачам. А викладачам дана система дозволяє вести електронний журнал обліку оцінок та відвідування, у будь-який момент оновлювати, налаштовувати матеріали до курсу та організовувати електронне обговорення актуальних проблем між студентами на форумах порталу. Викладачі та студенти мають змогу обмінюватися повідомленнями.

Доступ до ресурсів навчального порталу – персоніфікований. Логіном може слугувати пошта від Google або Microsoft або деяка інша. Адміністратор порталу допомагає реєструватися на навчальному порталі. Інструкції для реєстрації на порталі та інструкція для отримання доступу до електронних курсів розміщуються на сайті навчального порталу. Кожний користувач має доступ лише до тих електронних курсів, на які він був зареєстрований.

2.2 Структура електронного курсу

Електронний навчальний курс, що розміщується у межах інформаційного освітнього порталу, повинен мати уніфіковану структуру, а саме приведений до відповідної форми, тобто мати види діяльності для оцінювання студентів відповідно до дидактичної картки, яка розробляється у відповідності до робочої програми навчального курсу. Також у межах електронного курсу є доцільним відстежувати виконання елементів курсу та завершеності курсу. Структура електронного курсу включає в себе такі елементи:

1. Робоча програма, що затверджена в установленому порядку. Робоча програма містить мету та завдання вивчення курсу, зміст курсу, що відображає назви тем кожного модуля з анотаціями, кількістю годин (кредитів ECTS) на вивчення кожного модуля, форму підсумкового контролю (залік або іспит в усній або письмовій формі);
2. Тематичний план, який відображає план проведення лекційних та практичних (лабораторних або семінарських) занять, а також виконання учнями завдань для самостійної роботи;
3. Критерії оцінювання. Містить відомості щодо системи оцінювання поточних або підсумкових навчальних успіхів студентів з дисципліни. Для кожного модуля вказується розподіл балів за виконання завдань та шкала оцінювання. Наводиться таблиця співвідношень національних оцінок та оцінок ECTS [3].
4. Інтернет-джерела та Друковані. У цьому ресурсі пропонуються Інтернет-ресурси, основні та додаткові друковані джерела з дисципліни [3].
5. Глосарій (за потребою). Містить основні терміни електронного навчального курсу разом з їх визначенням.
6. Дидактична карта курсу. Містить коротку анотацію до курсу, відомості про автора (викладача лекційних та практичних занять) та його фото, розподіл

курсу за тижнями, темами і модулями, тематику практичних (семінарських, лабораторних) занять, розподіл балів за видами діяльності, позааудиторний контроль [3].

2.3 Структура модуля електронного курсу

Модуль курсу також має свою структуру. Модуль курсу містить наступні елементи:

1. Анотація до модуля. Містить цілі вивчення модуля, методичні рекомендації щодо вивчення основних матеріалів, карту модуля. Цілі модуля мають бути зрозумілими для учня, бути досяжними та конкретними.
2. Теоретичний навчальний матеріал. Може містити такі навчальні ресурси:
 - Опорні конспекти аудиторних лекцій;
 - Структуровані електронні матеріали, зміст яких відображає логіку навчання за курсом і надає студенту теоретичні відомості модуля у повному обсязі;
 - Мультимедійні презентації лекцій;
 - Додаткові електронні навчальні матеріали: гіперпосилання, флеш-ролики, аудіо і відео матеріали, електронні конспекти лекцій тощо [3];
 - Тести для самоконтролю.
3. Практичні, лабораторні, семінарські роботи. У матеріалах курсу окремими ресурсами мають міститися практичні, лабораторні та семінарські роботи. До кожної роботи має бути сформульовано мету та завдання, які забезпечують формування вмінь та навичок, методичні рекомендації з її виконання, форму подання результатів виконаної роботи, критерії оцінювання кожної роботи. Кожна робота повинна містити термін виконання.

4. Завдання для самостійної роботи. Вагома частина навчальних годин під час вивчення кожної дисципліни відводиться на самостійне опрацювання. Тому у матеріалах електронного навчального курсу необхідно розмістити додатковий теоретичний матеріал, завдання для самостійного виконання та методичний матеріал, який буде забезпечувати його якісне виконання студентами.
5. Модульний контроль. Для перевірки отриманих знань студентами використовуються індивідуальні завдання, тести та опитування [3].

3 ЗАСОБИ РОЗРОБКИ ЕЛЕКТРОННИХ КУРСІВ

3.1 Стандартні засоби

3.1.1 Moodle

Moodle – система управління навчанням, яка дозволяє створювати електронні курси в формі веб-сайту. Перші прототипи системи були створені в 1999 році Мартіном Дугіамасом (Martim Dougiamas), на той момент студентом університету Керттін у Австралії. Перший сайт на базі Moodle було створено у 2001 році, зараз число сайтів на Moodle перевищило за 1 мільйон по всьому світу. За цей час навколо Moodle сформувалося велике та сильне співтовариство, яке розробляє плагіни (модулі) і ділиться ними в мережі. За допомогою плагінів можна створити унікальну систему управління навчанням. Moodle встановлюється на сервері або розгортається в хмарі. Платформа добре підходить як для задач освітніх установ, навчальних центрів, так і для корпоративного навчання [4].

У Moodle можна завантажувати будь-який тип контенту: текстовий (включаючи PDF та XLS), зображення, презентації (через плагін Presentation), тести та курси. Відео можна завантажити без плагінів або для зручності підключити Medial – стрімінговий відеосервіс [4].

На смартфонах і планшетах Moodle можна відкривати в мобільних браузерях Chrome та Safari або використовувати застосунок Moodle Mobile [4].

З Moodle можна завантажувати будь-який вид звіту, але для цього потрібні плагіни [4].

Система є платною, мінімальна ціна в рік для 50 користувачів становить 120\$. Рішення у вигляді електронного курсу на Moodle не підходить з двох

причин: по-перше точно не відомо скільки користувачів буде для цього електронного курсу, по-друге матеріали електронного курсу – це конфіденційна інформація, яку ще не можна викладати в сторонній системі, через те, що книга на основі якої робиться цей курс ще не опублікована.

3.1.2 Moodle Cloud

Moodle Cloud – хмарна версія системи Moodle. Вона є безкоштовною, але безкоштовна версія також має ряд обмежень: максимально 50 користувачів і головне, якщо нею не користуватися протягом 60 днів, то електронний курс створений за допомогою неї автоматично видаляється. Таке рішення для створення електронного курсу в даній роботі також не підходить, бо невідомо коли почнуть користуватися електронним курсом плюс попередньо зазначені недоліки.

3.1.3 iSpring Suite

iSpring Suite – це конструктор у вигляді плагіна який працює з PowerPoint, який дозволяє створювати електронні курси в вигляді презентації. Недоліки цього інструмента в тому, що він коштує достатньо дорого, а саме більше 10000 грн у рік, а пробна версія доступна всього лише на 14 днів. Знов таки він не підходить з урахуванням високої ціни та попередньо перелічених недоліків.

3.1.4 iSpring Learn

iSpring Learn – платформа для онлайн-навчання, за допомогою якої можна створити учбовий портал, є аналогом платформи Moodle. Для того щоб запустити навчання в iSpring Learn не потрібно шукати сервер, встановлювати на нього систему і плагіни, а достатньо лише створити аккаунт, завантажити матеріали і назначити їх учням.

У iSpring Learn є можливість проводити лекції та вебінари. За словами розробників вебінари працюють на різних пристроях. Можна почати трансляцію з робочого комп'ютера, а продовжити з телефона на шляху на ділову зустріч. Усі записи зберігаються в HD-форматі.

За допомогою iSpring Learn можна створювати електронні курси, інтерактивні статті, тренажери, тести та монтувати відеолекції.

Ця платформа самостійно відкриває доступ до нових курсів, продовжує термін навчання і нагадує учням про дедлайн.

Платформа iSpring Learn збирає в одному місці результати тестування, прогрес по курсам і учбовим програмам. Можна оцінювати рівень підготовки кожного учня в реальному часі.

iSpring Learn працює на комп'ютерах, планшетах і телефонах. Учні можуть проходити курси навіть без доступу в Інтернет, наприклад в потягу або літаку.

Але ця платформа не підходить для створення електронного курсу в межах цієї роботи з огляду на свою ціну та невідомі у вигляді кількості учнів, точної дати виходу курсу і так далі.

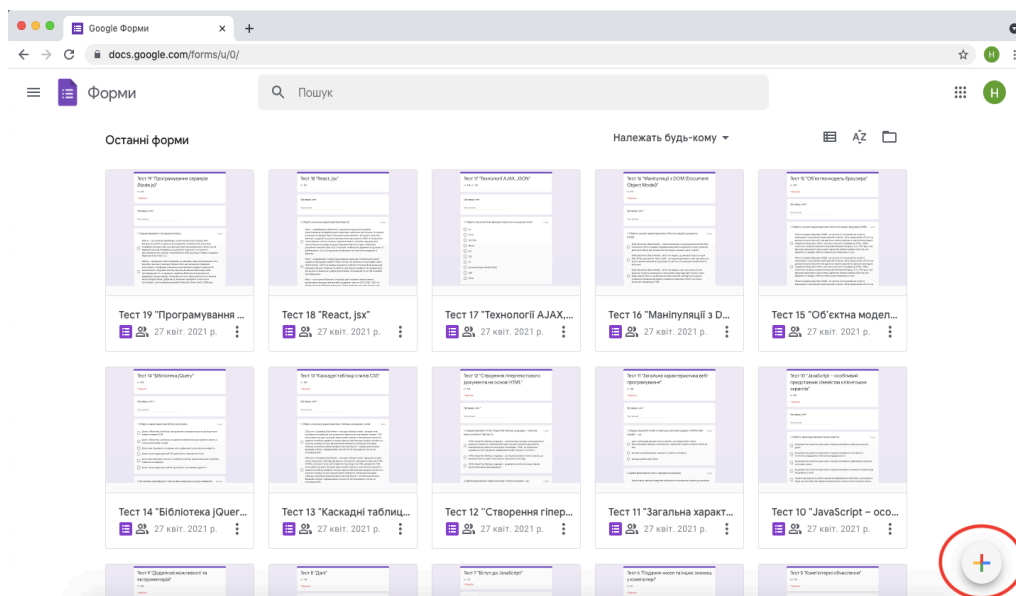
3.2 Google Forms

Google Forms – це безкоштовний додаток, який дозволяє створювати опитування, вікторини та тести. Google Forms дозволяють підібрати оформлення, збирати результати та статистику та доступні як з комп'ютерів так і з мобільних пристроїв. Google Forms дозволяють проводити опитування, складати список гостей, збирати електронні адреси для розсилки новин, а також проводити вікторини та складати тести. Google Forms було обрано у якості засобу для розробки тестів для модульного контролю в електронному курсі, створеного в межах цієї роботи.

Для того, щоб створити форму-тест потрібно:

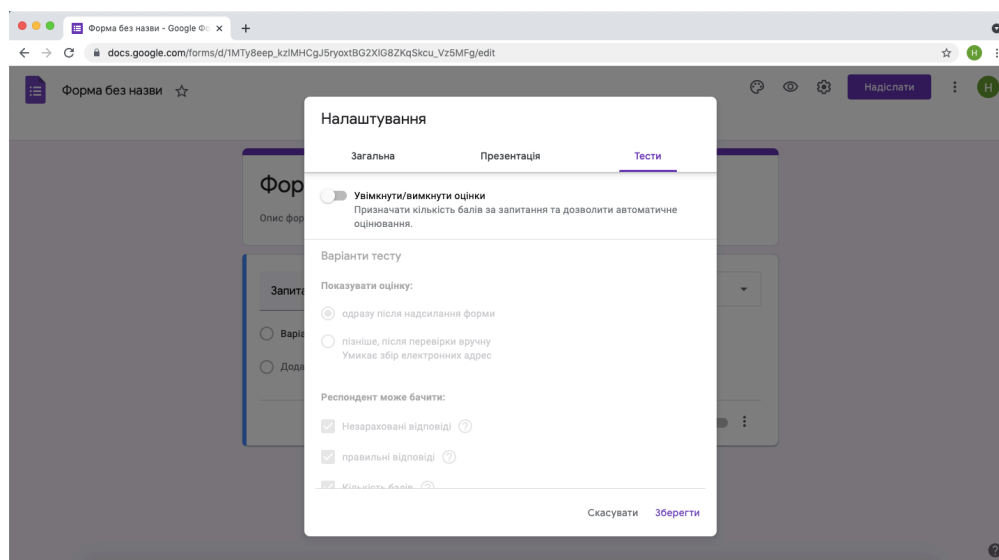
1. Мати Google аккаунт;
2. Обрати «Форми» у «Застосунки Google» справа, натиснувши на іконку з 6 крапок в Chrome;

3. Натиснути на кнопку «+», малюнок 3.1;



Малюнок 3.1

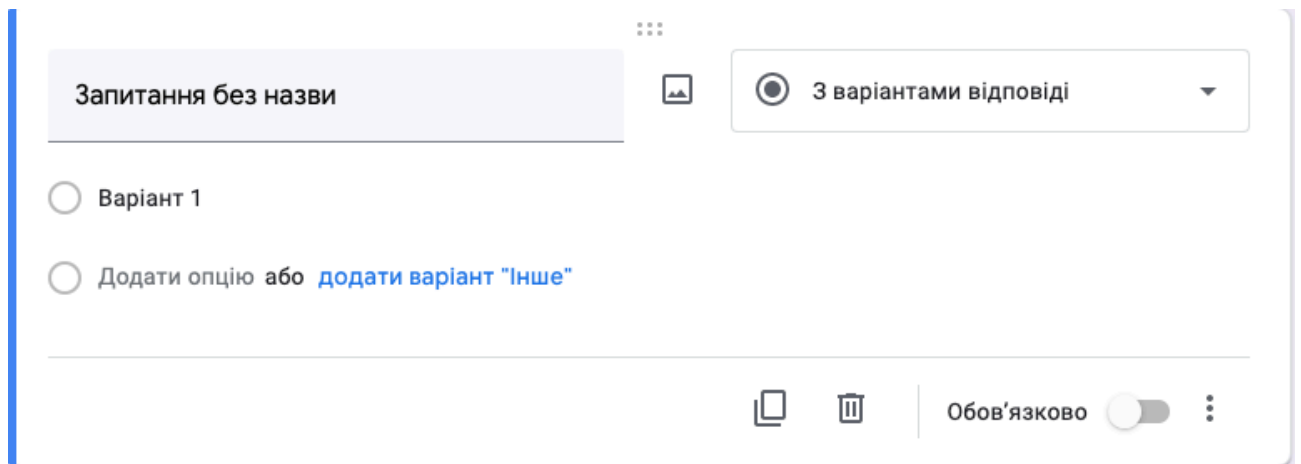
У вікні, що відкрилося, натиснути на іконку «Налаштування», перейти на вкладку «Тести» та увімкнути перемикач «Тест», малюнок 3.2.



Малюнок 3.2

Щоб створити запитання потрібно відкрити розділ «Запитання». Для запитань доступні такі формати:

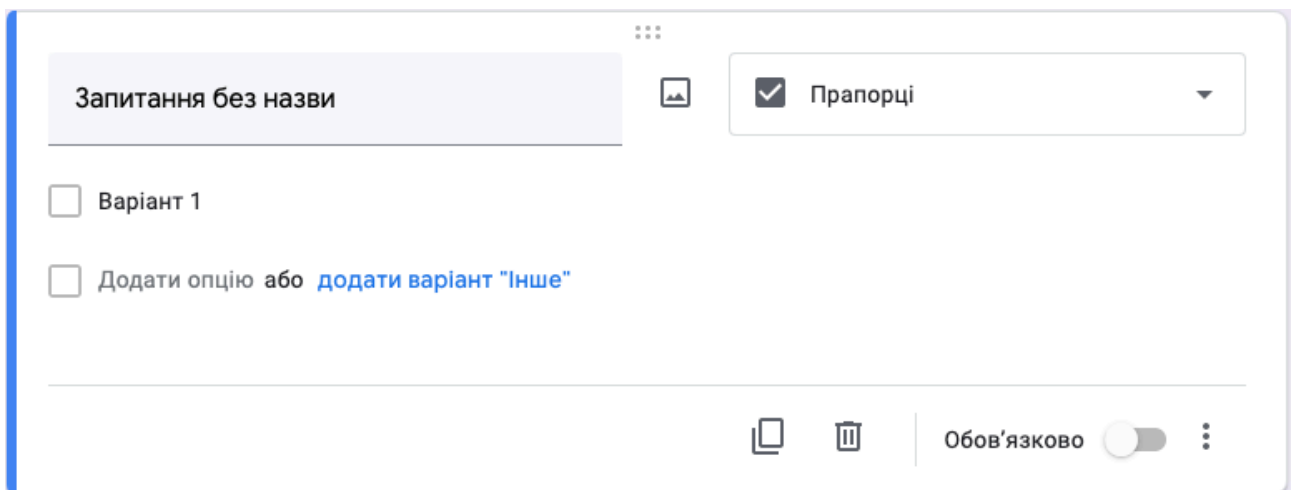
«3 варіантами відповіді» – для запитання, у якому лише одна відповідь правильна (Малюнок 3.3);



The screenshot shows a question editor interface. At the top, there is a header bar with a title input field containing "Запитання без назви" and a dropdown menu set to "3 варіантами відповіді". Below the header, there are two radio button options: "Варіант 1" and "Додати опцію або додати варіант 'Інше'". At the bottom right, there are icons for copy and delete, a toggle switch for "Обов'язково" (which is currently off), and a vertical ellipsis menu.

Малюнок 3.3

- «Прапорці» – для запитання, у якому в якості відповіді може бути декілька



The screenshot shows a question editor interface for a multiple-answer question. The header bar has the same title input field "Запитання без назви", but the dropdown menu is set to "Прапорці". Below the header, there are two checkbox options: "Варіант 1" and "Додати опцію або додати варіант 'Інше'". At the bottom right, there are icons for copy and delete, a toggle switch for "Обов'язково" (which is currently off), and a vertical ellipsis menu.

Малюнок 3.4

відповідей (Малюнок 3.4);

- «Таблиця з варіантами відповіді» – для запитання, у якому декільком запитанням потрібно встановити у відповідність відповіді з набору (малюнок 3.5).

Малюнок 3.5

Малюнок 3.6

Для кожного запитання потрібно обрати правильну відповідь та реакцію на неправильну відповідь (це опціонально і доступно не для всіх типів запитань (точно доступно для «3 варіантами відповіді» та «Прапорці»)). Реакція на неправильну відповідь – це інформація, яка показується користувачу, коли він обирає неправильну відповідь. Також є можливість встановити кількість балів за запитання. Усе це можна налаштувати,

натиснувши «Ключ опитування» внизу запитання, після цього з'явиться вікно для відповідних налаштувань (малюнок 3.6). Для користувача, у випадку неправильної відповіді, це буде виглядати так, як на рисунку 3.7.

- Запитання можна видаляти, редагувати. Можна змінювати позицію запитання, можна перемішувати запитання випадковим чином.

Для того, щоб пройти тест, користувачу не обов'язково мати Google аккаунт, йому просто достатньо перейти за посиланням, яке веде на Google тест, і йому відкриється Google форма з тестом. Після проходження тесту користувач бачить свої результати, а це: кількість зароблених балів, коментарі до неправильних відповідей.

Усі відповіді користувачів зберігаються в формі у розділі «Відповіді». Там також можна побачити статистику, яка включає: розподіл балів, середній рівень,

×

1. Охарактеризуйте основні відмінності між дискретним (цифровим) і 0/1 аналоговим принципом обробки інформації.

Для ілюстрації основної відмінності між дискретним (цифровим) і аналоговим принципом обробки інформації розглянемо класичний приклад з музичними інструментами. Рояль можна назвати дискретною системою, тому що в ній довільну фіксовану ноту можна або грати, або ні, бо кожна нота (клавіша) фізично (дискретно) відділена від наступної. Скрипка ж буде аналоговою системою, оскільки на скрипці висота тону може змінюватися неперервно, змінюючи положення смичка на струні. Дискретність можна уявити як утворення цегляної стінки будівлі із окремих цеглин, а неперервність характеризує водяний потік.

Для ілюстрації основної відмінності між дискретним (цифровим) і аналоговим принципом обробки інформації розглянемо класичний приклад з музичними інструментами. Скрипку можна назвати дискретною системою, тому що в ній довільну фіксовану ноту можна або грати, або ні, бо кожна нота фізично (дискретно) відділена від наступної. Рояль же буде аналоговою системою, оскільки на роялі висота тону може змінюватися неперервно. Неперервність можна уявити як утворення цегляної стінки будівлі із окремих цеглин, а дискретність характеризує водяний потік.

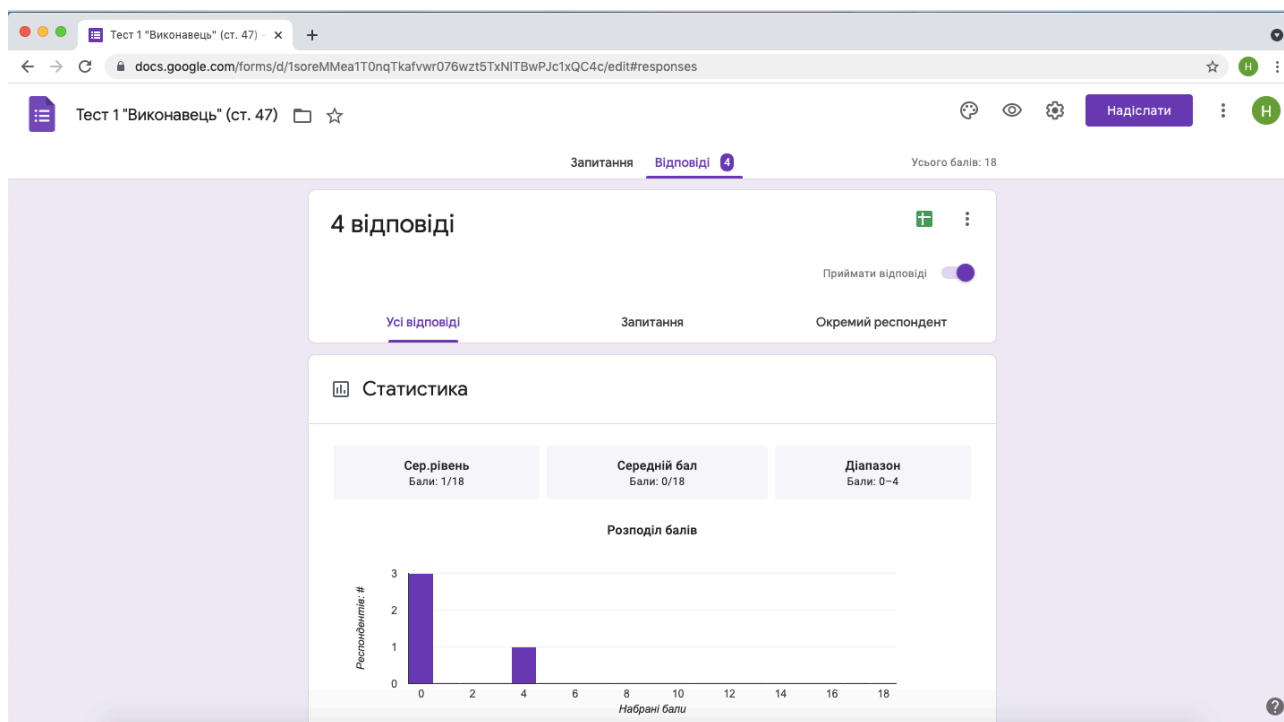
Між дискретним (цифровим) і аналоговим принципом обробки немає відмінностей.

Відгук

ст. 27 - 28

Малюнок 3.7

середній бал, діапазон набраних балів (малюнок 3.8), запитання на які часто відповідають неправильно. Також у цьому розділі можна переглянути відповіді окремих респондентів. Відповіді можна зберігати в Excel файлі.



Малюнок 3.8

У форму можна вставити: зображення, відео.

Цей інструмент є дуже зручним для створення тестів для модульного контролю з декількох причин: по-перше він повністю безкоштовний, по-друге тестові питання можна робити в різній формі, по-третє створені тести – це незалежні частини інформації, які можна поєднувати з іншою різною інформацією в різних формах (вбудувати на своєму веб-сайті, розмістити в системі управління навчанням, відправити електронною поштою і так далі).

4 ЕЛЕКТРОННИЙ КУРС «ДИТЯЧЕ ПРОГРАМУВАННЯ»

4.1 Призначення курсу

Електронний курс призначений для дітей віком 7 - 12 років. Також цей курс буде корисним для батьків, вчителів, які хочуть навчити дітей програмуванню. У електронному курсі розглядаються питання конструктивної С (опису як самих алгоритмів, так і обчислень, що ними задаються, тобто процесів послідовної обробки конструктивних об'єктів) в інформатиці [5]. У електронному курсі представлені базові підходи до побудови алгоритмів вирішення простих задач та їх реалізації на мові програмування. У електронному курсі викладено основні концепції програмування, які не потребують спеціальної математичної підготовки [5].

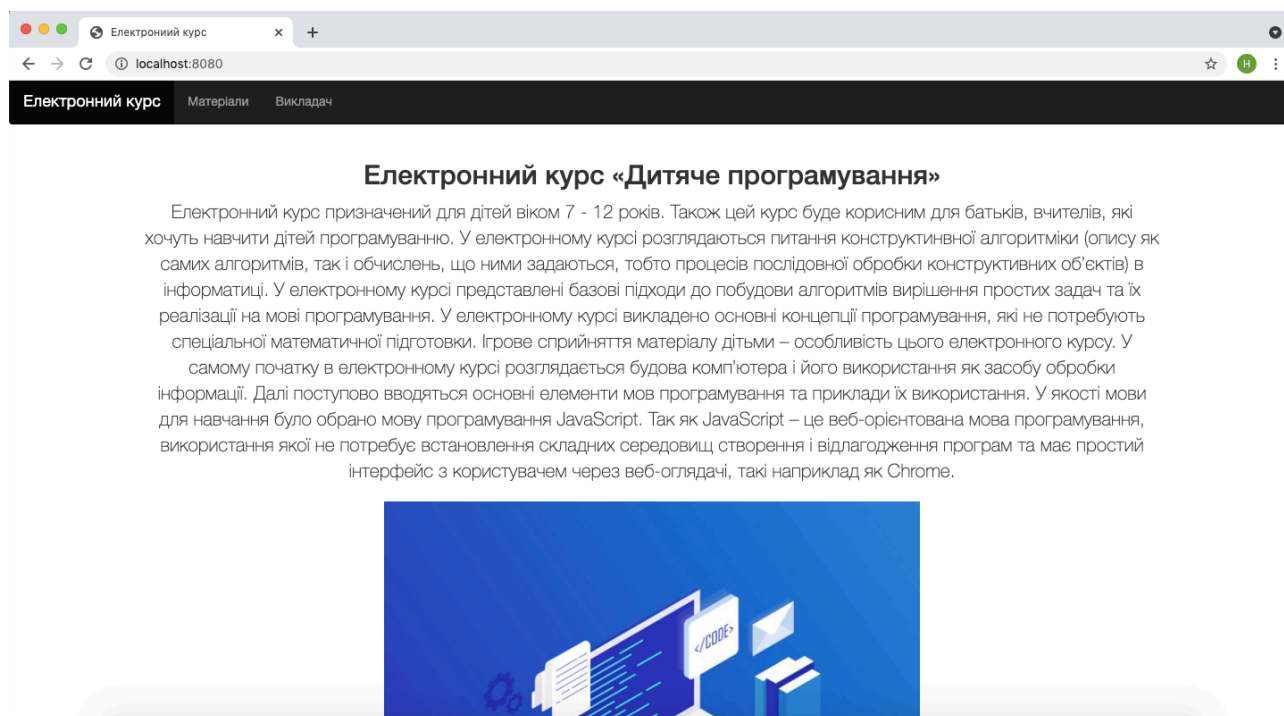
Ігрове сприйняття матеріалу дітьми – особливість цього електронного курсу. У самому початку в електронному курсі розглядається будова комп'ютера і його використання як засобу обробки інформації [5]. Далі поступово вводяться основні елементи мов програмування та приклади їх використання [5].

У якості мови для навчання було обрано мову програмування JavaScript. Так як JavaScript – це веб-орієнтована мова програмування, використання якої не потребує встановлення складних середовищ створення і відлагодження програм та має простий інтерфейс з користувачем через веб-оглядачі, такі наприклад як Chrome [5].

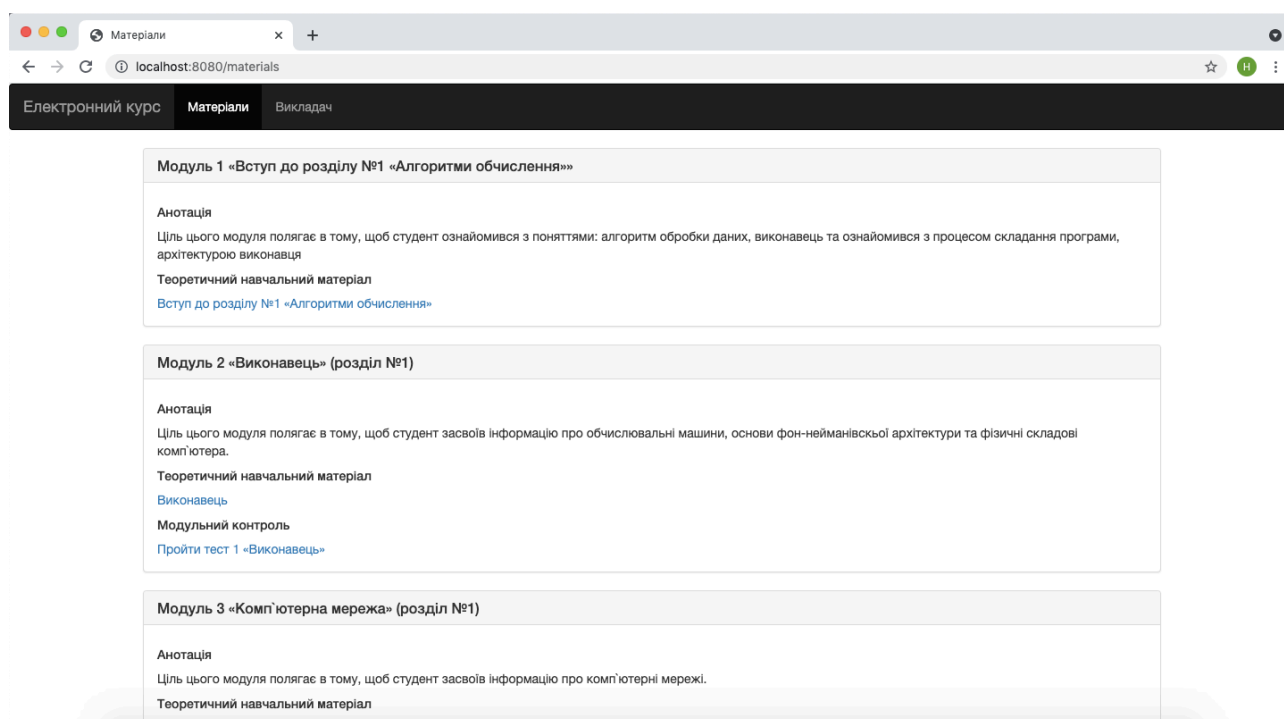
4.2 Зміст електронного курсу

4.2.1 Загальні відомості

Електронний курс повинен бути розміщений на веб-сайті, скріншот головної веб-сайту наведено на малюнку 4.1, який містить три сторінки: перша – це сторінка з назвою та призначенням курсу, друга з матеріалами та тестами до курсу зображено на рисунку 4.2, а третя – сторінка викладача.



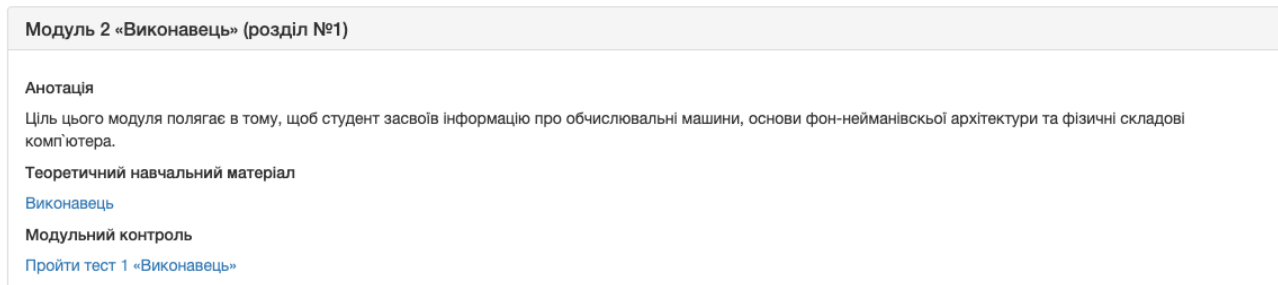
Малюнок 4.1



Малюнок 4.2

Теоретичний матеріал нарізано на невеликі частини у вигляді pdf файлів і взяті з книги М. М. Глибовця «Дитяче програмування». Питання в тестах також взяті з книги, але їхній формат, додаткові варіанти відповідей були розроблені

автором цієї роботи спираючись на матеріали книги та джерела в мережі Інтернет. Усі матеріали розділено на модулі (скріншот одного модуля, зображено на рисунку 4.3) так, як це зроблено за допомогою підрозділів у самій



Малюнок 4.3

книзі.

Кожен модуль складається з анотації, яка висловлює ціль модуля, частинки теоретичного навчального матеріалу в вигляді pdf файлу під назвою модуля та не завжди з тесту для контролю засвоєного матеріалу модуля (для модульного контролю) учнем. Модуль не завжди містить тест у якості модульного контролю, тому що модульний контроль може містити практичні задачі або взагалі не містити модульного контролю, бо сам модуль може бути вступом до розділу. Електронний курс складається з 23 модулів та 19 тестів.

Електронний курс, розроблений у межах цієї роботи, частково відповідає уніфікованому стандарту електронного курсу, бо містить лише модулі з теоретичним матеріалом та тестами, тобто основу електронного курсу, яка має бути доповнена контактною інформацією викладача, робочою програмою, тематичним планом і т. д. Даний електронний курс розбито на модулі згідно з сучасною тенденцією, яка полягає в мікронавчанні.

4.2.2 Тести для модульного контролю

Як вже раніше зазначалося, у якості інструмента для розробки тестів модульного контролю в електронному курсі було обрано Google Forms, тому під час розробки тестів були використані всі переваги цього інструмента.

Головна мета тестів модульного контролю – це допомогти учням засвоїти (повторити) отримані ними знання в процесі відпрацювання теоретичного навчального матеріалу, а також перевірити засвоєння цього матеріалу.

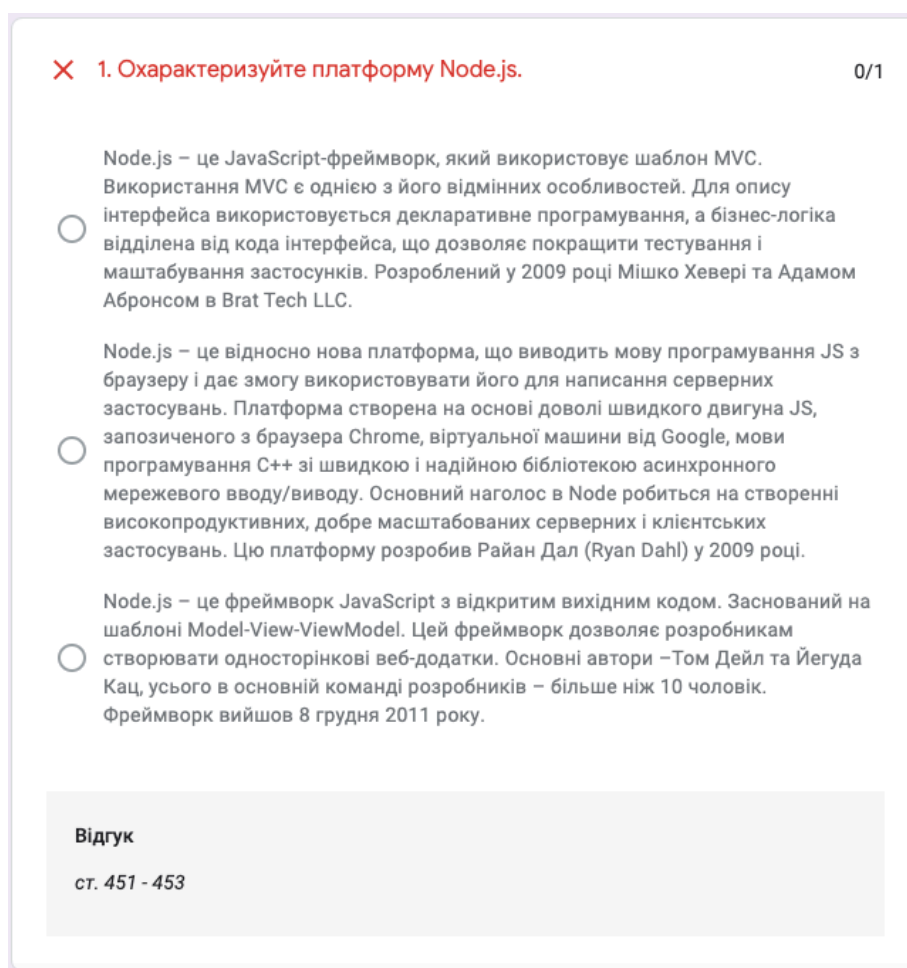
Кожен тест модульного контролю містить поле для вводу прізвища та імені учня і це поле є обов'язковим для заповнення. Для проходження тесту учню достатньо лише відкрити посилання з тестом і не потрібно вводити електронну gmail пошту або авторизуватися за допомогою неї. Біля назви тесту міститься сторінка з книги, звідки було взято основні запитання тесту (скріншот тесту, зображено на малюнку 4.4).

The screenshot shows a Google Form titled "Тест 19 \"Програмування серверів (Node.js)\"". Below the title, it says "ст. 473" and "*Обов'язкове поле". There is a text input field labeled "Прізвище, ім'я *". Below this, there is a question: "1. Охарактеризуйте платформу Node.js." with a score of "1 бал". The question text is: "Node.js – це JavaScript-фреймворк, який використовує шаблон MVC. Використання MVC є однією з його відмінних особливостей. Для опису інтерфейса використовується декларативне програмування, а бізнес-логіка відділена від кода інтерфейса, що дозволяє покращити тестування і масштабування застосунків. Розроблений у 2009 році Мішко Хевері та Адамом Абронсом в Brat Tech LLC." There are two radio button options: the first is "Node.js – це відносно нова платформа, що виводить мову програмування JS з браузера і дає змогу використовувати його для написання серверних застосунків. Платформа створена на основі доволі швидкого двигуна JS, запозиченого з браузера Chrome, віртуальної машини від Google, мови програмування C++ зі швидкою і надійною бібліотекою асинхронного мережевого вводу/виводу. Основний наголос в Node робиться на створенні високопродуктивних, добре масштабованих серверних і клієнтських".

Малюнок 4.4

Питання, які зустрічаються в тестах можуть бути декількох типів: з одним варіантом відповіді, з декількома варіантами відповіді та у формі сітки прапорців (для встановлення відповідності «слово – значення»).

Після проходження тесту учень бачить набрану ним кількість балів та питання, на які він відповів правильно чи ні. Для всіх питань, окрім питань у формі сітки прапорців для яких немає наступної функції, у випадку неправильної відповіді є відсилка до сторінки, на якій міститься правильна відповідь (зображено на малюнку 4.5).



Малюнок 4.5

Викладач в свою чергу також бачить відповіді учнів та усю статистику. Викладач може експортувати усі відповіді учнів у Excel файл.

Деякі тести для модульного контролю спираються на засвоєння учнем не лише на матеріали поточного модуля, а й попередніх. Питання можуть стосуватися визначення поняття, його опису і так далі

Мінімальна вартість тесту в балах становить 3 бали, а максимальна – 30. Розподіл балів за тести наведено в таблиці 4.1.

Таблиця 4.1

| Назва тесту | Вартість тесту в балах |
|---|------------------------|
| Тест 1 «Виконавець» | 18 |
| Тест 2 «Комп'ютерна мережа» | 10 |
| Тест 3 «Алгоритми» | 8 |
| Тест 4 «Програми» | 6 |
| Тест 5 «Комп'ютерні обчислення» | 14 |
| Тест 6 «Подання чисел та інших значень у комп'ютері» | 15 |
| Тест 7 «Подання чисел та інших значень у комп'ютері» | 12 |
| Тест 8 «Дані» | 30 |
| Тест 9 «Додаткові можливості та інструментарій» | 8 |
| Тест 10 «JavaScript - особливий представник сімейства клієнтських скриптів» | 10 |
| Тест 11 «Розділ №3. Веб-програмування» | 5 |
| Тест 12 «Створення гіпертекстового документа на основі HTML» | 20 |
| Тест 13 «Каскадні таблиці стилів CSS» | 19 |
| Тест 14 «Бібліотека jQuery» | 13 |
| Тест 15 «Об'єктна модель браузера» | 7 |
| Тест 16 «Маніпуляції з DOM (Document Object Model)» | 11 |

| Назва тесту | Вартість тесту в балах |
|--|------------------------|
| Тест 17 «Технології AJAX, JSON» | 11 |
| Тест 18 «React.jsx» | 3 |
| Тест 19 «Програмування серверів (Node.js)» | 17 |
| Усього балів: | 237 |

Якщо потрібно обмежити доступ до тестів, то можна створити окреме додаткове, обов'язкове запитання, яке буде потребувати кодового слова, яке буде відомо лише викладачу та учням.

4.2.3 Практичні завдання для модульного контролю

До двох модулів «Оператори управління» і «Функції» не передбачено модульного контролю у вигляді тестів, зате передбачено практичні завдання, які містять задачі. Кожна задача оцінюється певною кількістю балів. Максимальна кількість балів, якою оцінюється практичне завдання до модулю «Оператори управління» становить 32 бали, а – до модулю «Функції» – 47 балів. Усього практичні коштують $32 + 47 = 79$ балів.

4.2.4 Теоретичний навчальний матеріал

Найперший модуль присвячено вступу до розділу 1. Алгоритми та обчислення. У теоретичному навчальному матеріалі до модуля з'ясовується що собою представляє процес виконання програми. Процес виконання програми пояснюється на прикладі роботи пральної машини. У матеріалі висвітлюється поняття алгоритму обробки даних. У матеріалі пояснюється для чого пояснюється будь-яка форма запису алгоритму. Також написано з чого починається складання програми, вводиться поняття виконавця, архітектурі виконавця, дається визначення вхідній мові. У кінці розповідається про основні компоненти комп'ютера. Далі, у наступних модулях, уточнюються введені поняття.

Теоретичний навчальний матеріал до модуля «Виконавець» розпочинається з розповіді про обчислювальні машини, їхню історію, дається визначення обчислювальній машині. У матеріалі розповідається про цифрові, аналогові обчислювальні машини. Важливо те, що у матеріалі розгорнуто та на прикладі пояснюється різниця між дискретним (цифровим) і аналоговим принципом обробки інформації. У матеріалі ілюструється схема цифрової обчислювальної машини з поясненням для кожного компонента. Також у матеріалі розповідається про центральний процесор (з визначенням), графічний процесор, звуковий процесор, процесор цифрових сигналів. У матеріалі дається визначення нейрокомп'ютеру. У матеріалі розповідається про асемблери, компілятори та різницю між ними. Також розповідається про те, якими бувають комп'ютери, як зберігається інформація в пам'яті комп'ютера. Потім розповідається про суперкомп'ютери та персональні комп'ютери. У другій частині навчального теоретичного матеріалу розповідається про основи фон-нейманівської архітектури та пояснюються пов'язані з цією темою поняття. У третій частині матеріалі йдеться про фізичні складові комп'ютера: системний блок, материнську плату, процесор, жорсткий диск, твердотільний накопичувач (SSD), відеокарту, маніпулятор типу «миша», монітор та ще про деякі цікаві речі.

У теоретичному матеріалі до модуля «Комп'ютерні мережі» йдеться про комп'ютерні мережі, дається визначення комп'ютерній мережі, розповідається про класифікацію комп'ютерних мереж, про роботу WWW. Також у матеріалі дається визначення поняттю браузер, гіпертекст, поки стисло розповідається про HTML.

Теоретичний матеріал до модуля «Алгоритми» розпочинається розповіддю про умови виникнення науки теорії алгоритмів, про виникнення спеціалізованих обчислювальних пристроїв, про виникнення слова алгоритм, про три основні етапи розвитку теорії алгоритмів, загальні відомості про неї. Далі розповідається про поняття інформації, а також про основні три поняття в загальній інформатиці: задача, алгоритм, програма. Також далі детально

розглядається форма постановки задачі, а це є питання, яке може бути завданням або задачею. Далі розглядаються відмінності між задачею та завданням. Потім розповідається про поняття алгоритмічно нерозв'язної задачі. Після цього уточнюється поняття алгоритму. Потім на прикладі відомої задачі з човном, вовком, козою та капустою будується алгоритм її розв'язування. Також трохи згодом розповідається про блок-схеми. Більш розгорнутим прикладом побудови алгоритму, псевдо-коду, блок-схеми слугує задача про пірамідку. У кінці частини цього теоретичного матеріалу наводяться висновки, у яких зазначаються властивості будь-якого алгоритму.

Теоретичний матеріал до модуля «Програми» розпочинається тим, що зазначається необхідність наявності мови, яка буде описувати алгоритми виконавцям. Після цього за допомогою цитати А. П. Єршова описується яким має бути програміст. Потім наводяться визначення мови в загальному розумінні, мови в системах обробки інформації, алгоритмічної мови, командної мови, машинної мови. Далі наводиться визначення поняття «програма». Далі розповідається про те, яким має бути опис послідовності дій, призначений для автоматичного пристрою. Після цього наводиться ще одне визначення алгоритму, визначення алгоритмічної мови та програмування. Також розповідається про те, чим займається наука програмування. Після цього, перед тим, як перейти до прикладів програм, обговорюється питання про те, якими якостями повинна володіти хороша програма. Далі розповідається про програмування, про те, що таке парадигма програмування і про основні парадигми програмування. Потім розповідається про основні кроки розробки програми: уточнення способів організації даних – потрібних структур і типів даних, розробку алгоритму, запис алгоритму обраною мовою програмування, тестування, налагодження та документування програми.

Теоретичний матеріал до модуля «Комп'ютерні обчислення» розпочинається розповіддю про те, як комп'ютер виконує програму. Детально розповідається про роботу процесора, про його регістрову пам'ять, про машинні команди. Наводиться приклад роботи регістрів. Далі розповідається

про пам'ять комп'ютера. У матеріалі розповідається про поняття «порт», «драйвер», «файл» і «формат». Далі розповідається про машинну мову, мову високого рівня. Потім розповідається про трансляцію та інтерпретацію та різницю між ними.

У теоретичному матеріалі до модуля «Подання чисел та інших значень у комп'ютері» спочатку розповідається підрозділом про загальні принципи подання інформації. У цьому підрозділі йдеться про процес сприйняття інформації людиною. Після заглиблення в цей процес, читача плавно підводять до питання про представлення інформації у комп'ютері. У кінці цього підрозділу на прикладі розповідається про те, як за допомогою чисел можна закодувати слова. Наступний підрозділ матеріалу присвячено позиційним системам числення. Підрозділ розпочинається визначенням системи числення. Далі уточнюється яка саме система числення є позиційною. У підрозділі розповідається про те, що таке основа системи числення, про загальновживану форму запису числа і все це на прикладі десяткової системи числення. Далі у підрозділі розповідається про шістнадцяткову систему числення. Після цього детально пояснюється як переводити число з однієї системи числення в іншу на прикладі. Третій підрозділ присвячено двійковому кодуванню. Напочатку цього підрозділу зазначається, що інформація може бути закодована 0 і 1. Потім розповідається про те, що інформація представляється у вигляді повідомлень і яких саме. Після цього розповідається що таке кодування. Далі згадується що таке дані. Після цього зазначається якими можуть бути повідомлення, з чого вони складаються і те, що комп'ютер повинен вміти їх обробляти. Після цього розповідається про те, як перші комп'ютери відображали числа, використовуючи двійкову систему числення. Після цього детально розповідається як зберігаються дані, на прикладі чисел та надається визначення поняття «машинне слово». Після йдеться про кодування числової, текстової та графічної інформації.

У теоретичному матеріалі до модуля «Початки програмування» розповідається як навчитися програмувати якоюсь мовою програмування. Потім зазначається яку мову програмування обрано і чому саме.

Теоретичний матеріал до модуля «Вступ до JavaScript» розпочинається короткою характеристикою мови JavaScript, а саме підрозділом «Що таке JavaScript?» Тут розповідається про те, у якому році з'явилася мова JavaScript та для чого саме. Також зазначається у яких додатках вона використовується. Потім розповідається для чого мова JavaScript може використовуватися. Після цього розповідається чому мову JavaScript було названо саме так. Потім розповідається про характеристики мови JavaScript, а саме: що ця мова є динамічною, інтерпретованою мовою програмування з об'єктно-орієнтованими можливостями. Далі зазначається чому саме для програмування було обрано мову JavaScript. Потім, у кінці підрозділу, перелічуються переваги використання JavaScript. У наступному підрозділі коротко розповідається про засоби розробки JavaScript, а саме починаючи від засобів для початківців, такі як Word та браузері до просунутих, таких як Microsoft FrontPage, Macromedia HomeSite 5. Наступний підрозділ матеріалу присвячено початку роботи, а саме встановленню браузера, розгортанню консолі JavaScript. У наступному підрозділі «Перші кроки» пропонується розпочати з реалізації операції додавання – програми в один рядок та пояснюється робота інтерпретатора. Далі пропонується створити програму в декілька рядків та розповідається як це робити. Далі код програми розбирається порядково. Пояснюється як писати коментарі в мові JavaScript. Потім трохи розповідається про процедурне програмування. Після цього мова йде про те, що всі різновиди операторів зводяться до чотирьох основних: присвоювання, послідовної композиції, умовного оператора та умовного циклу. Далі розповідаються всілякі цікаві відомості про програми. Далі розповідається про семантику та синтаксис. Розповідається про вхідні дані, результат програми, про те, як все це зберігається в пам'яті. Потім пояснюється поняття змінної, розповідається про цикли з прикладом. Далі розповідається як алгоритм переробити на програмний

код. У останньому підрозділі матеріалу «Зауваження щодо синтаксису» йдеться про тонкощі синтаксису мови JavaScript.

Теоретичний матеріал до модуля «Дані» присвячено даним. У підрозділі «Сталі, змінні та типи даних» зазначається, що дані можна поділити на три великі групи: вхідні, проміжні і вихідні. Далі зазначається, що більшість програм потребують вхідних даних і розповідається про те, що в більшості випадків вхідні значення вводяться із зовнішніх носіїв або від людини за допомоги пристроїв вводу-виводу. Потім зазначається, що вводити значення з клавіатури не завжди зручно і тому використовуються псевдовипадкові числа. Далі розповідається про псевдовипадкові числа і як їх можна задати з прикладом програми. Потім у підрозділі нагадується як комп'ютер взаємодіє з оперативною пам'яттю під час виконання програми. Потім розповідається про зовнішню пам'ять, поняття «файл» та «формат». Потім у підрозділі розповідається про алфавіт. Далі розповідається про лексику мови, трохи про синтаксис мови JavaScript. Потім дається визначення сталим, іменам, зарезервованим (ключовим, службовим) словам, типу. Далі йдеться про висловлювання та визначення їх істинності. Потім розповідається про вирази та правила їх побудови. У кінці підрозділу розповідається про логічні формули. Наступний підрозділ матеріалу називається «Примітивні типи JavaScript». На початку цього підрозділу зазначається, що JavaScript – це мова з динамічною типізацією. Далі зазначається, що JavaScript дозволяє працювати з трьома примітивними типами даних: числами, рядками і логічними (булевими). Потім розповідається про null і undefined. Далі розповідається про змінні в JavaScript, основні операції над даними, оператори роботи з числами, спеціальні оператори роботи з рядками, спеціальні значення, масиви, об'єкти та символи і усе це супроводжують приклади.

Теоретичний матеріал до модуля «Оператори управління» присвячено операторам управління. Спершу розповідається, які оператори управління підтримує JavaScript. Далі у окремому підрозділі «Послідовне управління» розповідається про оператор послідовного управління у JavaScript. Зазначається, що в JavaScript роль послідовного оператора управління може брати на себе пустий оператор або «;». Наступний підрозділ «Групування» присвячено оператору групування. Зазначається, що цей оператор потрібен, коли потрібно декілька операторів об'єднати в групу. Наступний підрозділ «Розгалуження» присвячено оператору розгалуження (if, else). Цей оператор пояснюється досить детально (з прикладами). Наступний підрозділ «Варіант» присвячено оператору switch. Оператор вдало пояснюється на прикладі програми для формування розкладу за днями тижня. Для цього прикладу наводиться алгоритм і код програми. Наступний підрозділ «Ітерація» присвячено оператору ітерації. Спочатку розповідається про оператор while, потім про – do, while, потім про – for. Останній підрозділ присвячено інтерфейсам браузера: alert, prompt, confirm.

Теоретичний матеріал до модуля «Функції». Матеріал розпочинається з підрозділу «Парадигма процедурного програмування. Підпрограми». Підрозділ розпочинається з чотирьох принципів, описаних Тімом Пітерсом. Далі в підрозділі розповідається про те, що для того, щоб дотримуватися цих принципів потрібно використовувати декомпозицію і про саму декомпозицію. Далі мова йде про підпрограми та метод проектування зверху-донизу. Після детального знайомства з підпрограмами, зазначається що вони розділяються на дві різні категорії: процедури і функції. Далі дається визначення процедурам і функціям. Далі розповідається що таке функція в JavaScript. Підрозділ завершується визначенням процедурного програмування та розповіддю про нього. Наступний підрозділ називається «Визначення і виклик функцій». Підрозділ розпочинається з демонстрації синтаксису заголовку функції в JavaScript. Далі

розповідається про інструкцію «return». Далі автор пропонує написати функцію обчислення середнього арифметичного двох чисел. Далі розповідається про те, як функція викликається. Також далі наводиться приклад функції без інструкції return. Потім у підрозділі розповідається про функціональні літерали. Наступний підрозділ присвячено глобальним і локальним змінним. У цьому підрозділі розповідається що таке глобальні та локальні змінні. Наступний останній підрозділ називається матеріалу «Рекурсія», у ньому розповідається про рекурсію. У цьому підрозділі розповідається про числа Фібоначчі та про факторіал.

Теоретичний матеріал до модуля «Додаткові можливості та інструментарій» одразу розпочинається підрозділом, який називається «Інструментарій розробок». Підрозділ розпочинається поясненням що таке методологія проектування складних комп'ютерних систем. Далі розповідається про повторне використання і читача підводять до поняття «бібліотека програм». Після пояснення що таке бібліотека програм, описується що таке програмний фреймворк. Після цього дається визначення поняттю «застосунок». Далі знову йдеться про фреймворки. Другий останній підрозділ називається «Додаткові можливості». У ньому розповідається про екосистему JavaScript, фронтенд-розробку, застосунки командного рядка, розробку настільних додатків (GUI), розробку мобільних застосунків, розробку бекенду та автоматизацію застосувань за допомогою мови JavaScript.

Теоретичний матеріал до модуля «JavaScript – особливий представник родини клієнтських скриптів» розпочинається з визначення поняття «скрипт», якими мовами пишуться скрипти, визначенням поняття «скриптова мова». Далі описується сфера застосування скриптів. Далі йде підрозділ «Порівняння JavaScript з іншими мовами програмування». У цьому підрозділі зазначається, що JS має багато властивостей об'єктно-орієнтованої мови, але завдяки концепції прототипів вона все ж таки відрізняється від традиційних мов ООП [5]. Далі розповідається про прототипування програмного забезпечення. Потім порівнюється Java та JavaScript і зазначається, що в JS немає поняття класу, а

успадкування відбувається шляхом клонування існуючого примірника об'єкта – прототипу [5]. Далі на прикладі демонструється створення об'єкта на JS. Після цього детально, з прикладами розповідається про прототипування в JS, зокрема про вбудовані прототипи та про об'єкти, а також про фабрику об'єктів. Наступний підрозділ «Характеристика серверного JavaScript» досить стисло (порівняно з попередніми підрозділами) описує серверний JS. Останній підрозділ називається «Висновки» і він містить висновки щодо JavaScript. У цьому підрозділі описуються основні області застосування мови, переваги та недоліки мови та особливості мови, які роблять її унікальною та конкурентоспроможною.

У теоретичному матеріалі до модуля «Розділ 3. Веб-програмування» спочатку пояснюється робота архітектури «клієнт-сервер». Також пояснюється в чому полягає суть веб-програмування. Потім розповідається про різні мови веб-програмування та їх використання. Далі розповідається про те, які інтерактивні можливості взаємодії клієнта і сервера має оптимізувати розробник сайту. Після цього трохи розповідається про CSS. Потім розповідається про те, для чого були створені спеціалізовані мови сценаріїв, в тому числі і JavaScript. Далі обговорюється питання вартості програмного забезпечення і у цьому контексті згадуються PHP та MySQL. Після цього йдеться про те, якими інструментами має володіти хороший веб-програміст.

На початку теоретичного матеріалу до модуля «Створення гіпертекстового документа на основі HTML» дається визначення мові HTML і що таке гіпертекстовий документ. Трохи далі у матеріалі наводиться зразки гіпертексту. Далі пояснюється чим може бути гіперпосилання і як його активувати. Потім розповідається, що для створення HTML-документа можна скористатися текстовим редактором і пропонується завантажити обраний автором текстовий редактор. Далі пропонується створити першу «пробну» веб-

сторінку. Наступний підрозділ називається «Перша повноцінна сторінка». Спочатку розповідається про макет сторінки. Далі розповідається про елемент HTML та його компоненти, атрибути елементу, вкладені елементи та мета-теги. Далі розповідається про повноцінний HTML-документ та його складові. Наступний підрозділ носить назву «Становлення веб-майстра». У цьому підрозділі розповідається про редагування тексту, а саме про атрибути для цього, колір, стиль і шрифт тексту. Потім розповідається про рисунки, таблиці, посилання, рухомий рядок, заголовок `<!DOCTYPE>`, створення графіки за допомогою елемента HTML Canvas. У останньому підрозділі наводиться приклад створення календаря, годинника, м'ячика, що рухається по екрану.

Теоретичний матеріал до модуля «Каскадні таблиці стилів CSS» розпочинається підрозділом «Вступ до CSS», який в свою чергу розпочинається визначенням CSS. Далі розповідається як і чому з'явився CSS, як він застосовується до сторінки. Далі розповідається що таке селектор та розглядаються їх можливі види. Після цього у матеріалі розглядається зв'язок між CSS та HTML, інакше кажучи, як до HTML можна підключити CSS. Далі розглядаються правила і каскадність CSS, одиниці виміру в CSS, робота з кольором і фоном, робота зі шрифтом. Далі йдеться про особливості синтаксису CSS. Після цього розпочинається підрозділ присвячений селекторам, у якому розповідається про селектори типу, селектори класів, селектори за ID, універсальний селектор, комбінатори, селектори за атрибутами, псевдокласи, комбінування селекторів.

Теоретичний матеріал до модуля «Бібліотека jQuery» розпочинається підрозділом, що називається «Загальна характеристика», у якому наводиться загальна характеристика бібліотеки jQuery. У цьому ж підрозділі надається визначення терміну «плагін». Також у цьому підрозділі розповідається звідки

завантажити та як підключити jQuery. Далі мова йде про селектори і на прикладах розглядаються варіанти доступу до елементів. Далі розповідається про оброблювач готовності документу, де розповідається про два способи оброблення. Далі в підрозділі розповідається про створення елементів DOM за допомогою jQuery. Потім розповідається про роботу з отриманим набором значень, маніпулювання об'єктами на сторінці, підключення обробників, співпрацю jQuery з AJAX. У кінці в окремому підрозділі детально розповідається як розробити гру «П'ятнашки» з використанням jQuery.

Теоретичний матеріал до модуля «Об'єктна модель браузера» присвячено об'єктній моделі браузера. Матеріал розпочинається розповіддю про те, що JavaScript використовується не лише для роботи з веб-браузером. Далі розповідається що таке об'єктна модель браузера та для чого вона використовується. Далі розповідається про місце ОБМ у структурі JS. Далі розповідається про основні об'єкти об'єктної моделі браузера і особливості деяких з них (наприклад, window) розглядаються більш детально ніж інших.

Теоретичний матеріал до модуля «Маніпуляції з DOM (Document Object Model)» починається з підрозділу, який називається «Основна характеристика DOM», який в свою чергу починається поясненням що таке DOM (Document Object Model). Також пояснюється, що для DOM дерева існує два типи вузлів, яким чином можна задавати корінь дерева і доступатися до інших вузлів дерева. Далі наводиться приклад маніпуляції з DOM за допомогою JS. Далі пояснюється чим є документ згідно з DOM-моделлю. Далі приклади DOM-документів та допоміжні рисунки. Далі розглядається приклад складнішого документа з атрибутами плюс малюнок з деревом тегів. Далі у підрозділі пояснюється навіщо потрібна ієрархічна модель DOM. Трохи далі докладніше розбираються способи доступу та властивості елементів DOM. Після цього розпочинається підрозділ,

який носить назву «Властивості й атрибути, пошук», який присвячено властивостям й атрибутам DOM-елементів та пошуком за ними. Наступний підрозділ називається «Зміна сторінки через DOM». У цьому підрозділі розглядається як написати HTML, CSS і JS код для повідомлення, що виводиться посередині екрану та закривається кнопкою «ОК», яка також міститься на 9 екрані. Створення та закриття повідомлення потрібно реалізувати за допомогою засобів маніпуляції DOM у JS [5].

Теоретичний матеріал до модуля «Технології AJAX, JSON» починається загальною характеристикою AJAX. У цій характеристиці дається визначення технології AJAX, наводиться список технологій які використовуються в концепції AJAX. Далі пояснюється як працює AJAX на прикладі роботи з Facebook через інструменти розробника в браузері. У наступному підрозділі, що називається «Обмін даними з сервером» розповідається про об'єкт XMLHttpRequest, а також про методи GET та POST. Далі розповідається про методи XMLHttpRequest, після цього розповідається про використання об'єкта XMLHttpRequest. Далі йдеться про співпрацю AJAX з jQuery. У кінці розділу робляться підсумки, у яких сформовано базові кроки використання AJAX та додаткова інформація для більш глибокого опанування AJAX. Наступний підрозділ присвячено JSON і він починається його загальною характеристикою, далі розповідається про синтаксис JSON, інтерпретацію JSON.parse(), серіалізацію JSON.stringify(), створення об'єктів за допомогою JSON, симулювання асоціативних масивів, проведення ітерацій JSON та наводяться приклади застосування.

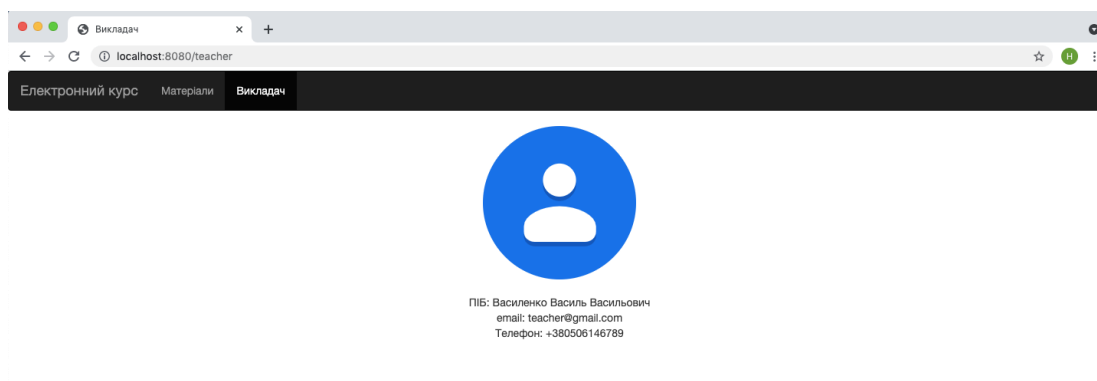
Теоретичний матеріал до модуля «React.jsx» починається з пояснення що таке React та опису його особливостей. Трохи далі надається визначення поняттю «рєндеринг, комп'ютерна візуалізація (англ. rendering)». Потім

описуються основні відмінності React від інших подібних по застосуванню фреймворків і бібліотек. Далі у матеріалі описується як розпочати роботу з React. Після цього для ілюстрації роботи з бібліотекою розглядається така задача: створити клас, в якому викликається відмальовка кнопки з монетами [5]. На прикладі цієї задачі розглядаються тонкощі створення програм за допомогою React.

Теоретичний матеріал до модуля «Програмування серверів (Node.js)» присвячено Node.js. Матеріал розпочинається підрозділом «Загальна характеристика», у якому розповідається що таке Node.js, хто і коли її розробив, що лежить в основі Node.js та що дозволяє робити Node.js. Наступний підрозділ «Технологія Node.js» розповідає про те, що може бути цікавого в Node.js як технології спочатку трохи розповідається про мову програмування JS, про основний недолік JS та як і з ним впорається Node.js. Також у підрозділі розповідається для чого призначений Node.js. Наступний підрозділ називається «Архітектура: потоки або асинхронний ввід/вивід з управлінням подіями» і присвячено потокам та паралелізму в Node.js. Наступний підрозділ носить назву «Продуктивність та використання процесора» розповідається про те, наскільки продуктивний Node.js з прикладом тесту. Наступний підрозділ присвячено основним модулям Node.js, таким як: file system, HTTP і т.д. Також у підрозділі розповідається про потоки та створення простого файлового сервера.

4.3 Сторінка викладача

На сайті передбачена сторінка з інформацією про викладача. Інформація вигадана, вона повинна бути замінена справжньою. Скріншот сторінки викладача зображено на рисунку 4.6.



Малюнок 4.6

4.4 Інструкція для вивчення електронного курсу

Електронний курс можна вивчати як з відвідуванням лекцій, так і самостійно, бо навчальний теоретичний матеріал подано досить розгорнуто, зрозуміло та з прикладами, а тести перевіряються автоматично. До електронного курсу повинен бути прикріплений викладач, який буде стежити за дедлайнами виконання завдань, за успішністю учнів, шляхом аналізу зібраної аналітики та буде відповідати на запитання учнів.

Для того, щоб освоїти матеріали електронного курсу, потрібно послідовно засвоювати навчальні теоретичні матеріали до модулів, потім після засвоювання навчального теоретичного матеріалу кожного модуля, виконувати модульний контроль у вигляді тесту або практичних задач.

5 ПРОГРАМНА РЕАЛІЗАЦІЯ ЕЛЕКТРОННОГО КУРСУ

5.1 Обраний інструментарій

5.1.1. Мова Java

5.1.1.1 Загальний огляд

Java – об'єктно-орієнтована мова програмування, випущена 1995 року компанією «Zune Microsystems» як основний компонент платформи Java [6]. Java програма компілюється у байт-код, який під час виконання інтерпретується віртуальною машиною для конкретної платформи. Саме Java була обрана основною мовою програмування для створення веб-сайту.

5.1.1.2 Встановлення на комп'ютер

З офіційного сайту <https://java.com/en/download/> можна встановити Java або якщо у Вас Mac OS, то можна встановити Java через пакетний менеджер Homebrew. На комп'ютері має з'явитися папка з JDK. Для того, щоб перевірити чи встановлена Java на комп'ютері з операційною системою Windows або Mac OS потрібно в терміналі ввести команду `java -version`. Якщо Java встановлена, то ця команда виведе версію Java.

5.1.1.3 Обрана версія Java

Для розробки веб-сайту було обрано 11 версію Java.

5.1.2 Spring Boot

У якості фреймворку для розробки було обрано Spring Boot. Spring Boot – це веб-фреймворк Java з відкритим кодом, що базується на мікросервісах. Структура Spring Boot створює повністю готове до виробництва середовище, яке повністю налаштовується за допомогою його попередньо побудованого коду в межах своєї кодової бази. Архітектура мікросервісу надає розробникам повністю закриту програму, включаючи вбудовані сервери додатків [7].

5.1.2.1 Огляд Spring Boot

Spring Boot – це лише розширення вже існуючих та розширюваних фреймворків Spring, але він має деякі специфічні особливості, які полегшують роботу програми в екосистемі розробника. Це розширення включає заздалегідь

налаштування веб-початкові набори, які допомагають полегшити відповідність сервера додатків, необхідну для інших проектів Spring [7].

5.1.2.2 Переваги Spring Boot

Spring Boot має ряд функцій, які роблять його чудовим для швидкої розробки програм Java, включаючи автоматичну конфігурацію, перевірку працездатності та надійні залежності. Фітчі та їхні переваги наведено в таблиці 5.1.

Таблиця 5.1

| Фітча | Перевага |
|--|---|
| Автономний додаток | Є можливість просто створити jar-програму та запустити її, без потреби налаштування розгортання [7]. |
| Вбудовані сервери | Поставляється з попередньо вбудованими серверами додатків Tomcat, Jetty та Undertow, які не потребують подальшого встановлення для використання. Це також забезпечує більш швидкі та ефективніші розгортання, що призводить до скорочення часу перезапуску [7]. |
| Автоконфігурування | Spring та інші сторонні фреймворки будуть налаштовані автоматично [7]. |
| Фітчі подібні до production властивостей | Перевірки працездатності, показники та зовнішні конфігурації [7]. |

| Фітча | Перевага |
|---------------------|--|
| Залежності стартера | Це забезпечує надійні залежності, розроблені для спрощення конфігурації збірки. Це також забезпечує повну гнучкість інструменту збірки (Maven і Gradle) [7]. |

5.1.2.3 Як працює Spring Boot?

Автоматичні конфігурації зводиться до трьох простих анотацій Spring Boot:

- `@SpringBootApplication`
- `@EnableAutoConfiguration`
- `@ComponentScan`

`@SpringBootApplication` використовується у вхідній точці застосунку, щоб додати клас, у якому він знаходиться, до основного методу програми [7].

`@EnableAutoConfiguration` надає кожному представницькому класу можливість автоматичної конфігурації [7].

Нарешті, `@ComponentScan` при ініціалізації сканує всі компоненти та декларації пакета [7].

Spring Boot включає не тільки анотації, а також використовує Spring Starter Dependencies, щоб забезпечити запуск додатка з правильними залежностями [7].

У багатьох випадках, коли додаток стає більшим, важко правильно налаштувати залежності проекту, плагіни Spring Boot Starter допоможе

полегшити управління залежностями. Прикладом залежностей Spring Starter є веб-залежність Spring Boot Starter [7].

5.1.3 IntelliJ IDEA

5.1.3.1 Загальний огляд

IntelliJ IDEA – інтегроване середовище розробки для різних мов програмування (Java, Python, Scala, PHP та інших) [8]. Середовище доступне у двох версіях: «Community Edition» (урізана версія) та «Ultimate Edition». Доступна під Linux, Mac OS і Windows. Саме це інтегроване середовище розробки (урізана версія) використовувалося для розробки прототипу веб-сайту для електронного курсу.

5.1.3.2 Встановлення на комп'ютер

Для того, щоб встановити IntelliJ IDEA потрібно перейти на сайт <https://www.jetbrains.com/ru-ru/idea/download/>.

5.1.4 Thymeleaf

Для створення одиниць, які повторюються на сайті використовувався шаблонізатор Thymeleaf.

Thymeleaf – це сучасний механізм для створення шаблонів на Java на стороні сервера, як для веб так і для десктопних, мобільних середовищ [9]. Thymeleaf можна легко підключити у проєкті Spring Boot, для цього потрібно додати відповідні налаштування до файлу pom.xml.

HTML шаблони написані за допомогою Thymeleaf виглядають та працюють так само як HTML, що дозволяє їм працювати в застосунках, як корисним артефактам дизайну.

5.1.5 HTML

HTML (англ. HyperText Markup Language) – це мова розмітки, яка використовується для того, щоб сповістити браузер, як відображати веб сторінки, які відвідуються. HTML складається з ряду елементів, які використовуються, для того щоб вкладати та обертати різні частини контенту,

щоб змусити контент відображатися або діяти певним чином. Теги, що оточують, можуть зробити слово або зображення посиланням на щось ще, можуть зробити слова курсивом, зробити шрифт більше або менше і так далі. HTML семантично описує структуру веб сторінки.

Елементи HTML – це будівельні блоки сторінок HTML. Завдяки конструкціям HTML, об'єкти можуть бути вбудовані у візуальну сторінку. HTML задає засоби для створення структурованих документів, позначаючи структурну семантику тексту, наприклад заголовки, абзаци, списки, посилання та інші елементи [10]. Елементи HTML окреслюються тегами, що пишуться за допомогою кутових дужок. Теги не відображаються браузером, але використовуються для інтерпретації вмісту сторінки.

У HTML можна вбудовувати програми, написані на мові сценаріїв, наприклад JavaScript, що впливає на поведінку та вміст веб сторінок. Включення CSS визначає компоновання вмісту і вигляд. World Wide Web Consortium (W3C), який супроводжує стандарти HTML та CSS заохочує використовувати CSS над явним презентаційним.

Сторінки веб-сайту зроблено за допомогою HTML.

5.1.6 Bootstrap

5.1.6.1 Загальний огляд

Bootstrap – це безкоштовний набір інструментів (фреймворк) з відкритим кодом, призначений для створення веб-сайтів та вебдодатків, який містить шаблони CSS та HTML для типографіки, форм, кнопок, навігації та інших компонентів інтерфейсу, а також додаткові розширення JavaScript [11].

Bootstrap є таким популярним через те, що він дозволяє верстати сайти в декілька разів швидше, ніж на чистому CSS і JavaScript. Інший аспект – доступність. Вона зводиться до того, що надає можливість навіть веб розробнику-початківцю (без глибоких знань і достатньої практики) створювати достатньо якісні макети.

5.1.6.2 Підключення

Щоб використовувати Bootstrap його просто достатньо підключити до сторінки. Після цього стануть доступні інструменти даного фреймворку: стовпчикова система (сітка Bootstrap), класи і компоненти.

5.1.6.3 З чого складається Bootstrap?

Bootstrap складається з:

- Інструментів для створення макета (обгорткових контейнерів, потужної системи сіток, гнучких медіа-об'єктів, адаптивних утилітних класів) [12];
- Класів для стилізації базового контенту: тексту, зображень, коду, таблиць і figure [12];
- Готових компонентів: кнопок, форм, горизонтальних і вертикальних навігаційних панелей, слайдерів, випадаючих списків, акордеонів, модальних вікон, спливаючих підказок і іншого [12];
- Утилітних класів для вирішення традиційних завдань, які найбільш часто виникають перед веб-розробниками: вирівнювання тексту, відображення та приховування елементів, завдання кольору, фону, margin і padding відступів, і так далі [12].

5.1.6.4 Обрана версія Bootstrap

Була обрана версія Bootstrap 3.4.1.

5.1.7 Apache Maven

Apache Maven – фреймворк для автоматизованої збірки проектів на основі опису їх структури в файлах на мові POM (англ. Project Object Model), що є підмножиною XML. Проект Maven видається співтовариством Apache Software Foundation, де формально є частиною Jakarta Project. Назва системи є словом з мови ідиш, сенс якого можна приблизно висловити як «збирач знань» [13].

Maven забезпечує декларативну, а не імперативну (на відміну від засобу автоматизації збирання Apache Ant) збірку проекту. У файлах опису проекту

міститься його специфікація, а не окремі команди виконання. Всі завдання по обробці файлів, описані в специфікації, Maven виконує за допомогою їх обробки послідовністю вбудованих і зовнішніх плагінів.

Maven використовується для побудови і управління проектами, написаними на Java, C #, Ruby, Scala, та іншими мовами.

Серед примітних альтернатив - система автоматичного складання Gradle, побудована на принципах Apache Ant і Maven, але використовує спеціалізований DSL на Groovy замість POM-конфігурації.

Саме Maven було обрано у якості засобу збірки для проекту вебсайту, розробленого у межах цієї роботи.

5.1.8 Apache Tomcat

Apache Tomcat - контейнер сервлетів, розроблений Apache Software Foundation. Повністю написаний мовою програмування Java та реалізує специфікацію сервлетів и Java Server Pages від Sun Microsystems, що є стандартами для розробки веб-застосунків на Java [14].

5.1.9 jQuery

jQuery – набір функцій JavaScript, що фокусується на взаємодії JavaScript і HTML. Бібліотека jQuery допомагає легко отримувати доступ до будь-якого елементу DOM, звертатися до атрибутів і вмісту елементів DOM, маніпулювати ними. Також бібліотека jQuery надає зручний API для роботи з AJAX. Розробка jQuery ведеться командою добровольців на пожертвування. Обрана версія 3.5.1.

5.1.10 JavaScript

JavaScript – мова програмування, яка дозволяє створювати динамічно оновлюємий контент, керувати мультимедіа, анімувати зображення. Мова підтримує об'єктно-орієнтований, імперативний і функціональні стилі. Є реалізацією специфікації ECMAScript (стандарт ECMA-262) [15].

Основні архітектурні риси JavaScript: динамічна типізація, слабка типізація, автоматичне управління пам'яттю та прототипне програмування [15].

5.2 Особливості програмної реалізації

5.2.1 Створення початкового коду для веб-сайту

Початковий код для вебсайту, розробленого в межах цієї роботи, не створювався з нуля. Щоб його отримати, потрібно перейти на сайт <https://start.spring.io/> та обрати необхідні налаштування (у моєму випадку Maven, Java і так далі) та натиснути «Generate» і на комп'ютер завантажиться архів з проектом, який потрібно розархівувати та відкрити в IntelliJ IDEA або в іншому середовищі розробки. У розархівованому проекті вже створені початкові класи та ієрархія папок.

5.2.1 Модель-Вигляд-Контролер

Фреймворк Spring Boot дозволяє реалізувати шаблон Модель-Вигляд-Контролер (MVC, Model-view-controller). Модель-Вигляд-Контролер – це архітектурний шаблон, який використовується під час проектування та розробки програмного забезпечення [16]. Цей шаблон дозволяє розділити систему на три окремі, але взаємопов'язані частини: модель даних, вигляд (інтерфейс користувача) та модуль керування (контролер). Таке розділення системи на окремі частини дозволяє незалежно змінювати їх окремо одну від одної (наприклад, частину інтерфейсу від частини для роботи з даними). Саме цей шаблон було використано в реалізації прототипу веб-сайту.

5.3 Основні компоненти

OnlineCourseApplication – це клас, який запускає застосунок, містить метод main та код запуску застосунку.

Клас MainController, який є контролером і відповідає за повернення сторінок за url. Повертає головну сторінку та матеріали. За головну сторінку відповідає html файл з назвою home, а за матеріали – materials.

home.html потрібний для відображення основної інформації про електронний курс, а materials.html містить усі модулі курсу з тестами та матеріалами. teacher.html містить інформацію про викладача.

У проєкті є фрагменти на Thymeleaf, а саме `header.html`, який відповідає за верхнє меню і вставляється на сторінках `home` та `materials`, `materialItem.html`, `materialItemWithTest.html` та `materialItemWithTasks.html` які відповідають за відображення одного модулю (без тесту, з тестом, з практичними). Фрагменти `materialItem.html` та `materialItemWithTest.html` містять змінну для назви модулю, змінну для анотації, змінну з посиланням на матеріал та `materialItemWithTest.html` містить змінну з посиланням на тест.

Скрипт `menuClick.js` відповідає за правильне відображення обраних користувачем пунктів меню.

Код, описаний вище, знаходиться у Додатку А.

Матеріали зберігаються в pdf файлах, кожен матеріал у окремому файлі. Практичні завдання також зберігаються кожне в окремому файлі pdf.

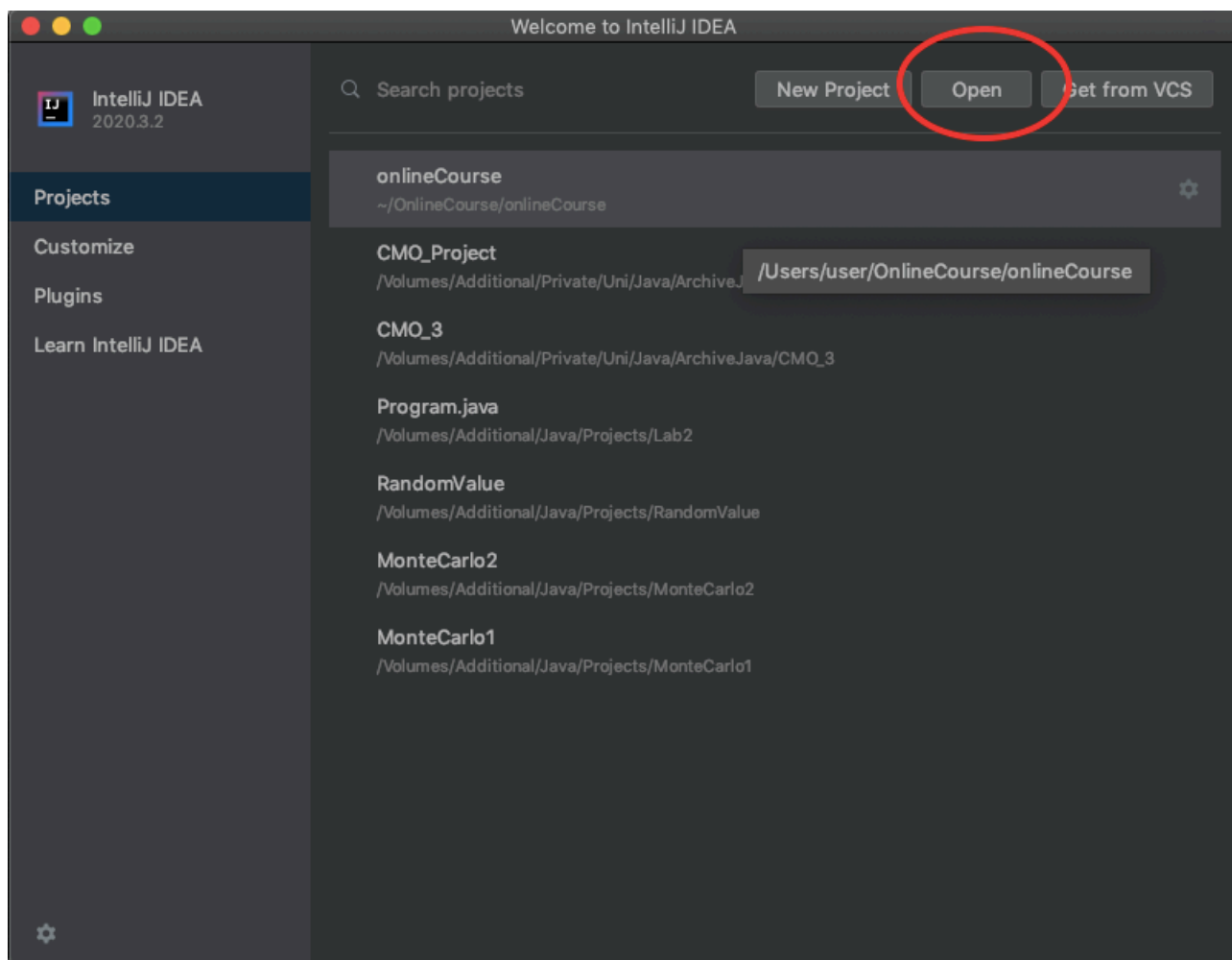
5.4 Інструкція з локального розгортання проєкту з веб-сайтом

Для того, щоб розгорнути проєкт необхідно, щоб на комп'ютері були встановлені: Java 11 версії та IntelliJ IDEA. Як їх встановити описано в підрозділах раніше. Далі потрібно розархівувати архів з проєктом. Після цього відкрити IntelliJ IDEA. У IntelliJ IDEA обрати «Open» та обрати проєкт `onlineCourse`, як на рисунку 5.1.

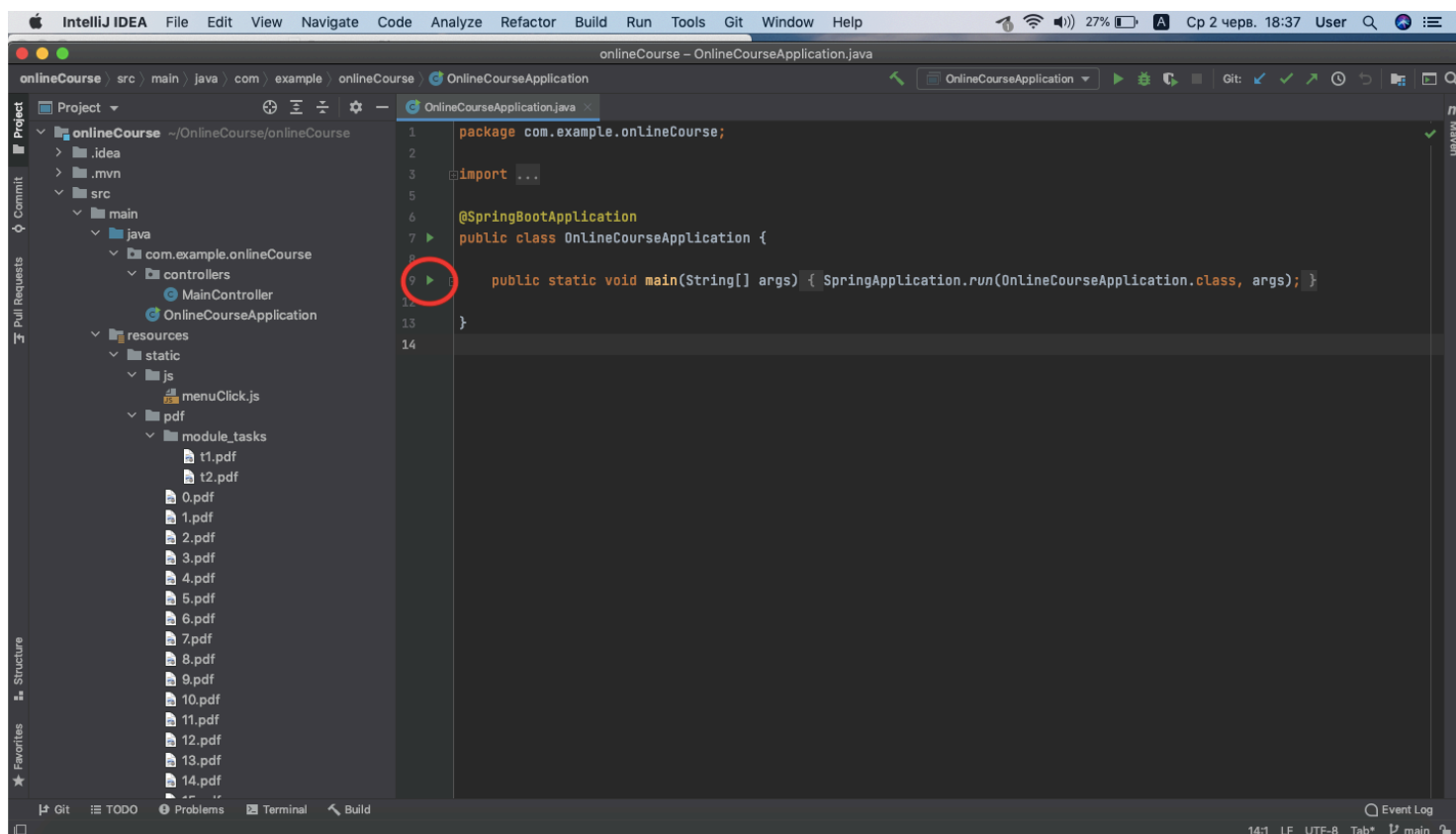
Після цієї дії проєкт має підвантажити необхідні залежності та розгорнутися без помилок, як показано на рисунку 5.2. Потрібно відкрити файл `OnlineCourseApplication.java`. Далі потрібно натиснути зелену кнопку запуску, яка обведена червоним на рисунку 5.2.

Далі в терміналі мають з'явитися логи, як показано на рисунку 5.3.

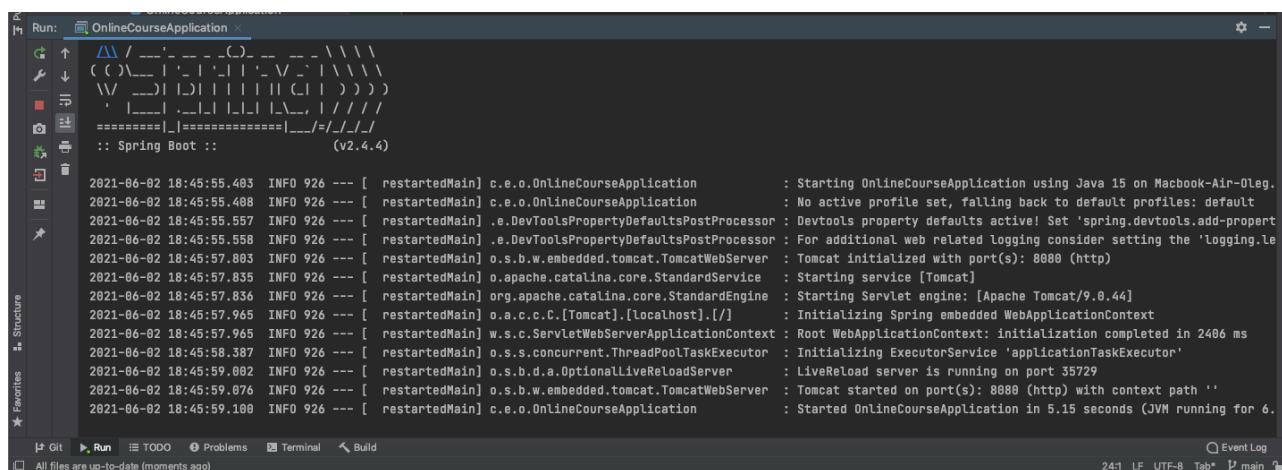
Після цього сайт доступний за посиланням <http://localhost:8080/>.



Малюнок 5.1



Малюнок 5.2



Малюнок 5.3

ВИСНОВКИ

У межах цієї роботи було:

- Досліджено програмні засоби розробки електронних курсів.
- Визначено особливості вивчення та методики представлення нюансів мови JavaScript у електронному курсі «Дитяче програмування» .
- Побудовано набір текстових запитань та вірних відповідей на них
- Програмно реалізовано тести з 4 типами питань (з одним варіантом відповіді, з декількома варіантами відповіді, таблиця з варіантами відповіді, з короткою відповіддю).
- Обрано локальний варіант розміщення електронного курсу «Дитяче програмування» через конфіденційність інформації.
- Для розміщення електронного курсу створено прототип сайту.

Використані джерела

1. <https://www.guide2research.com/research/lms-elearning-statistics>
2. <https://sike.ru/e-learning-modnyj-trend-ili-obrazovanie-budushchego>
3. <http://surl.li/rdrq>
4. <https://lmslist.ru/free-sdo/obzor-moodle/>
5. «Вступ до дитячого програмування. Мова JavaScript» М.М. Глибовець
6. <https://uk.wikipedia.org/wiki/Java>
7. <https://www.jrebel.com/blog/what-is-spring-boot> <http://surl.li/rxau>
8. https://uk.wikipedia.org/wiki/IntelliJ_IDEA
9. <https://www.thymeleaf.org/>
10. <https://uk.wikipedia.org/wiki/HTML>
11. <https://uk.wikipedia.org/wiki/Bootstrap>
12. <https://itchief.ru/bootstrap/introduction>
13. https://ru.wikipedia.org/wiki/Apache_Maven
14. https://ru.wikipedia.org/wiki/Apache_Tomcat
15. <https://ru.wikipedia.org/wiki/JavaScript>
16. <http://surl.li/rxau>

Додаток А

OnlineCourseApplication.java:

```
package com.example.onlineCourse;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
public class OnlineCourseApplication {

    public static void main(String[] args) {
        SpringApplication.run(OnlineCourseApplication.class, args);
    }

}
```

MainController.java:

```
package com.example.onlineCourse.controllers;

import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.GetMapping;

@Controller
public class MainController {

    @GetMapping("/")
    public String home(Model model) {
        model.addAttribute("title", "Електронний курс");
        return "home";
    }

    @GetMapping("/materials")
    public String materials(Model model) {
        model.addAttribute("title", "Матеріали");
        return "materials";
    }

    @GetMapping("/teacher")
    public String teacher(Model model) {
```

```

    model.addAttribute("title", "Викладач");
    return "teacher";
}
}

```

home.html:

```

<!DOCTYPE HTML>
<html xmlns:th="http://www.thymeleaf.org">
<head>
    <title th:text="${title}" />
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/
bootstrap.min.css">
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/
bootstrap.min.js"></script>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></
script>
</head>
<body>
<header th:insert="blocks/header :: header"></header>
<div class="container">
    <div class="py-5 text-center">
        <h2>Електронний курс «Дитяче програмування»</h2>
        <p class="lead">

```

Електронний курс призначений для дітей віком 7 - 12 років. Також цей курс буде корисним для батьків, вчителів, які хочуть навчити дітей програмуванню. У електронному курсі розглядаються питання конструктивної алгоритміки (опису як самих алгоритмів, так і обчислень, що ними задаються, тобто процесів послідовної обробки конструктивних об'єктів) в інформатиці.

У електронному курсі представлені базові підходи до побудови алгоритмів вирішення простих задач та їх реалізації на мові програмування. У електронному курсі викладено основні концепції програмування, які не потребують спеціальної математичної підготовки.

Ігрове сприйняття матеріалу дітьми – особливість цього електронного курсу. У самому початку в електронному курсі розглядається будова комп'ютера і його використання як засобу обробки інформації. Далі поступово вводяться основні елементи мов програмування та приклади їх використання.

У якості мови для навчання було обрано мову програмування JavaScript. Так як JavaScript – це веб-орієнтована мова програмування, використання якої не потребує встановлення складних середовищ створення і відлагодження програм та має простий інтерфейс з користувачем через веб-оглядачі, такі наприклад як Chrome.

```

    </p>
    
    <!--
    <div class="hero-unit">
        <h1>Heading</h1>
        <p>Tagline</p>
        <p>
            <a class="btn btn-primary btn-large">
                Learn more
            </a>
        </p>
    </div>
    -->
</div>
</div>
</body>
</html>

```

materials.html:

```

<!DOCTYPE HTML>
<html xmlns:th="http://www.thymeleaf.org">
<head>
    <title th:text="${title}" />
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <!-- https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css -->
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/
bootstrap.min.css">
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/
bootstrap.min.js"></script>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></
script>
</head>
<body>
<header th:insert="blocks/header :: header"></header>

<div class="container">

    <div th:insert="blocks/materialItem :: materialItem(
        'Модуль 1 «Вступ до розділу №1 «Алгоритми обчислення»»',

```

'Ціль цього модуля полягає в тому, щоб студент ознайомився з поняттями: алгоритм обробки даних,

виконавець та ознайомився з процесом складання програми, архітектурою виконавця',

'Вступ до розділу №1 «Алгоритми обчислення»',

'pdf/0.pdf')">

</div>

<div th:insert="blocks/materialItemWithTest :: materialItemWithTest(

'Модуль 2 «Виконавець» (розділ №1)',

'Ціль цього модуля полягає в тому, щоб студент засвоїв інформацію про обчислювальні машини,

основи фон-нейманівської архітектури та фізичні складові комп'ютера.',

'Пройти тест 1 «Виконавець»',

'https://docs.google.com/forms/d/

1soreMMea1T0nqTkafvwr076wzt5TxNITBwPJc1xQC4c/edit?usp=sharing',

'Виконавець',

'pdf/1.pdf')">

</div>

<div th:insert="blocks/materialItemWithTest :: materialItemWithTest(

'Модуль 3 «Комп'ютерна мережа» (розділ №1)',

'Ціль цього модуля полягає в тому, щоб студент засвоїв інформацію про комп'ютерні мережі.',

'Пройти тест 2 «Комп'ютерна мережа»',

'https://docs.google.com/forms/d/1LtEs6nBVTpr2flCyzcs_IoFXY9-wXTUo-UXfG3qpoKE/edit?usp=sharing',

'Комп'ютерна мережа',

'pdf/2.pdf')">

</div>

<div th:insert="blocks/materialItemWithTest :: materialItemWithTest(

'Модуль 4 «Алгоритми» (розділ №1)',

'Ціль цього модуля полягає в тому, щоб студент познайомився з поняттями алгоритму, інформації, задачею, завданням, навчився будувати алгоритми, блок-схеми.',

'Пройти тест 3 «Алгоритми»',

'https://docs.google.com/forms/d/

1qU35c0aa1lQxy8OfRD7Awj6w1cMXV6S1NEYELv43VOA/edit?usp=sharing',

'Алгоритми',

'pdf/3.pdf')">

</div>

<div th:insert="blocks/materialItemWithTest :: materialItemWithTest(

'Модуль 5 «Програми» (розділ №1)',

'Ціль цього модуля полягає в тому, щоб студент познайомився з поняттями програми, алгоритмічної, машинної мови.',

'Пройти тест 4 «Програми»',

'[https://docs.google.com/forms/d/](https://docs.google.com/forms/d/1CJLYC41WWQ3Hz9b9V8pATIo8BsrnEXqE4ZsqnEwAcF8/edit?usp=sharing)

1CJLYC41WWQ3Hz9b9V8pATIo8BsrnEXqE4ZsqnEwAcF8/edit?usp=sharing',

'Програми',

'pdf/4.pdf')">

</div>

<div th:insert="blocks/materialItemWithTest :: materialItemWithTest(

'Модуль 6 «Комп'ютерні обчислення» (розділ №1)',

'Ціль цього модуля полягає в тому, щоб студент познайомився з процесом комп'ютерних обчислень.',

'Пройти тест 5 «Комп'ютерні обчислення»',

'[https://docs.google.com/forms/d/](https://docs.google.com/forms/d/1lSWn2G3oPA8JG70zqHpv8dOnS9d5bOSekGpb5pXGdcc/edit?usp=sharing)

1lSWn2G3oPA8JG70zqHpv8dOnS9d5bOSekGpb5pXGdcc/edit?usp=sharing',

'Комп'ютерні обчислення',

'pdf/5.pdf')">

</div>

<div th:insert="blocks/materialItemWithTest :: materialItemWithTest(

'Модуль 7 «Подання чисел та інших значень у комп'ютері» (розділ №1)',

'Ціль цього модуля полягає в тому, щоб студент сформував уявлення про подання чисел та інших значень у комп'ютерах.',

'Пройти тест 6 «Подання чисел та інших значень у комп'ютері»',

'[https://docs.google.com/forms/d/](https://docs.google.com/forms/d/1l5E6CKIyTeWO5__z5RvbHr8ZXxxfATdwELe9KmahQU/edit?usp=sharing)

1l5E6CKIyTeWO5__z5RvbHr8ZXxxfATdwELe9KmahQU/edit?usp=sharing',

'Подання чисел та інших значень у комп'ютері',

'pdf/6.pdf')">

</div>

<div th:insert="blocks/materialItem :: materialItem(

'Модуль 8 «Вступ до розділу №2 «Початки програмування»»',

'Ціль цього модуля полягає в тому, щоб студент почав писати програми мовою JavaScript.',

'Вступ до розділу №2 «Початки програмування»',

'pdf/7.pdf')">

</div>

<div th:insert="blocks/materialItemWithTest :: materialItemWithTest(

'Модуль 9 «Вступ до JavaScript» (розділ №2)',

'Ціль цього модуля полягає в тому, щоб студент сформував уявлення про мову JavaScript.',

'Пройти тест 7 «Подання чисел та інших значень у комп'ютері»',

```
'https://docs.google.com/forms/d/
1Kzg608jlB9QqMqG-4-7gScnocQi6RILSu4RDePayDqM/edit?usp=sharing',
'Вступ до JavaScript',
'pdf/8.pdf')">
</div>
```

```
<div th:insert="blocks/materialItemWithTest :: materialItemWithTest(
'Модуль 10 «Дані» (розділ №2)',
'Ціль цього модуля полягає в тому, щоб студент сформував уявлення про дані
в мові JavaScript.',
'Пройти тест 8 «Дані»',
'https://docs.google.com/forms/d/1A6J8K5clSZNS1LPkQaye5pGSb5e-
yD5eW5VP3CTKq9s/edit?usp=sharing',
'Дані',
'pdf/9.pdf')">
</div>
```

```
<div th:insert="blocks/materialItemWithTasks :: materialItemWithTasks(
'Модуль 11 «Оператори управління» (розділ №2)',
'Ціль цього модуля полягає в тому, щоб студент сформував уявлення про
оператори управління в JavaScript.',
'Оператори управління',
'pdf/10.pdf',
'pdf/module_tasks/t1.pdf',
'Практичне завдання 1'
)">
</div>
```

```
<div th:insert="blocks/materialItemWithTasks :: materialItemWithTasks(
'Модуль 12 «Функції» (розділ №2)',
'Ціль цього модуля полягає в тому, щоб студент сформував уявлення про
функції в мові JavaScript.',
'Функції',
'pdf/11.pdf',
'pdf/module_tasks/t2.pdf',
'Практичне завдання 2'
)">
</div>
```

```
<div th:insert="blocks/materialItemWithTest :: materialItemWithTest(
'Модуль 13 «Додаткові можливості та інструментарій» (розділ №2)',
'Ціль цього модуля полягає в тому, щоб студент сформував уявлення про
бібліотеки, фреймворки, розробку фронкенду, бекенду,
маобільних застосунків на мові JavaScript',
'Пройти тест 9 «Додаткові можливості та інструментарій»',
```

'https://docs.google.com/forms/d/10OieA5HpOK-7YmApGomYyGea6XWxzCnDHxwyG0TJkk/edit?usp=sharing',
 'Додаткові можливості та інструментарій',
 'pdf/12.pdf')">
 </div>

<div th:insert="blocks/materialItemWithTest :: materialItemWithTest(
 'Модуль 14 «JavaScript – особливий представник сімейства клієнтських скриптів» (розділ №2)',
 'Ціль цього модуля полягає в тому, щоб студент дізнався про те, що таке скрипт, скриптові мови, про порівняння JavaScript з іншими мовами, про прототипування в JavaScript, про характеристику серверного JavaScript.',
 'Пройти тест 10 «JavaScript – особливий представник сімейства клієнтських скриптів»',
 'https://docs.google.com/forms/d/1yoA3pPyB1epvuscWCELzqGqs3PfTF2KBIEI-AUmIJl/edit?usp=sharing',
 'JavaScript – особливий представник сімейства клієнтських скриптів',
 'pdf/13.pdf')">
 </div>

<div th:insert="blocks/materialItemWithTest :: materialItemWithTest(
 'Модуль 15 «Розділ №3. Веб-програмування»',
 'Ціль цього модуля полягає в тому, щоб студент ознайомився з загальною характеристикою JavaScript.',
 'Пройти тест 11 «Розділ №3. Веб-програмування»',
 'https://docs.google.com/forms/d/1ltf10JsOCFQlGZmegLKFTgnfAhzwwgWUM4z-oXAPzzpE/edit?usp=sharing',
 'Розділ №3. Веб-програмування',
 'pdf/14.pdf')">
 </div>

<div th:insert="blocks/materialItemWithTest :: materialItemWithTest(
 'Модуль 16 «Створення гіпертекстового документа на основі HTML» (розділ №3)',
 'Ціль цього модуля полягає в тому, щоб студент опанував мову HTML та навчився створювати веб-сторінки.',
 'Пройти тест 12 «Створення гіпертекстового документа на основі HTML»',
 'https://docs.google.com/forms/d/19iMQTURtvr6TIvMK8UpJFvq-LKlSMVmWjhH4nEi5gmM/edit?usp=sharing',
 'Створення гіпертекстового документа на основі HTML',
 'pdf/15.pdf')">
 </div>

<div th:insert="blocks/materialItemWithTest :: materialItemWithTest(

'Модуль 17 «Каскадні таблиці стилів CSS» (розділ №3)',

'Ціль цього модуля полягає в тому, щоб студент опанував каскадні таблиці стилів CSS.',

'Пройти тест 13 «Каскадні таблиці стилів CSS»',

'[https://docs.google.com/forms/d/](https://docs.google.com/forms/d/1UUIg8p7dD1bpzAHSSdMqDBtTbgmLN_9BLKWxNWn5oA/edit?usp=sharing)

1UUIg8p7dD1bpzAHSSdMqDBtTbgmLN_9BLKWxNWn5oA/edit?usp=sharing',

'Каскадні таблиці стилів CSS',

'pdf/16.pdf')">

</div>

<div th:insert="blocks/materialItemWithTest :: materialItemWithTest(

'Модуль 18 «Бібліотека jQuery» (розділ №3)',

'Ціль цього модуля полягає в тому, щоб студент познайомився з бібліотекою jQuery.',

'Пройти тест 14 «Бібліотека jQuery»',

'[https://docs.google.com/forms/d/1VhF2Pg6wPCAEWbJtFn--](https://docs.google.com/forms/d/1VhF2Pg6wPCAEWbJtFn--rgsk7b4_qrgrjc5rs3MFDso/edit?usp=sharing)

rgsk7b4_qrgrjc5rs3MFDso/edit?usp=sharing',

'Бібліотека jQuery',

'pdf/17.pdf')">

</div>

<div th:insert="blocks/materialItemWithTest :: materialItemWithTest(

'Модуль 19 «Об'єктна модель браузера» (розділ №4)',

'Ціль цього модуля полягає в тому, щоб студент познайомився з об'єктною моделлю браузера.',

'Пройти тест 15 «Об'єктна модель браузера»',

'[https://docs.google.com/forms/d/](https://docs.google.com/forms/d/1cJ5oBcyduKGztgUXPWde8WsAQqEBnXosCBCaZY-N79A/edit?usp=sharing)

1cJ5oBcyduKGztgUXPWde8WsAQqEBnXosCBCaZY-N79A/edit?usp=sharing',

'Об'єктна модель браузера',

'pdf/18.pdf')">

</div>

<div th:insert="blocks/materialItemWithTest :: materialItemWithTest(

'Модуль 20 «Маніпуляції з DOM (Document Object Model)» (розділ №4)',

'Ціль цього модуля полягає в тому, щоб студент створив уявлення про DOM та маніпуляцію з ним.',

'Пройти тест 16 «Маніпуляції з DOM (Document Object Model)»',

'[https://docs.google.com/forms/d/](https://docs.google.com/forms/d/1Uef5Ud6i8u3Bu6v3oIXGamBTDCWFYbNxKHwFzM6YKs0/edit?usp=sharing)

1Uef5Ud6i8u3Bu6v3oIXGamBTDCWFYbNxKHwFzM6YKs0/edit?usp=sharing',

'Маніпуляції з DOM (Document Object Model)',

'pdf/20.pdf')">

</div>

<div th:insert="blocks/materialItemWithTest :: materialItemWithTest(

'Модуль 21 «Технології AJAX, JSON» (розділ №4)',

'Ціль цього модуля полягає в тому, щоб студент створив уявлення про технології AJAX та JSON.',

'Пройти тест 17 «Технології AJAX, JSON»',

'https://docs.google.com/forms/d/

1yWySka8NqqFzAQ9dotKTkHQEl3GIN_tPpHiNMLyXrTU/edit?usp=sharing',

'Технології AJAX, JSON',

'pdf/21.pdf')">

</div>

<div th:insert="blocks/materialItemWithTest :: materialItemWithTest(

'Модуль 22 «React, jsx» (розділ №4)',

'Ціль цього модуля полягає в тому, щоб студент створив уявлення про React, jsx.',

'Пройти тест 18 «React, jsx»',

'https://docs.google.com/forms/d/1MtUDbiHh8YisaC3xNO3lUCB-

xga4LJyejOrqBCdq_PQ/edit?usp=sharing',

'React, jsx',

'pdf/22.pdf')">

</div>

<div th:insert="blocks/materialItemWithTest :: materialItemWithTest(

'Модуль 23 «Програмування серверів (Node.js)» (розділ №4)',

'Ціль цього модуля полягає в тому, щоб студент створив уявлення про програмування серверів (Node.js).',

'Пройти тест 19 «Програмування серверів (Node.js)»',

'https://docs.google.com/forms/d/19vUBjqbq_N-

NnkJgkJuRCzvDCvY_g6ZjN3EozEOIPHA/edit?usp=sharing',

'Програмування серверів (Node.js)',

'pdf/23.pdf')">

</div>

</div>

</body>

</html>

teacher.html:

<head>

<title th:text="{title}" />

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>

```

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
</head>
<body>
<header th:insert="blocks/header :: header"></header>

<div class="container">
  <div class="py-5 text-center">
    
    <p>
      <br>
      ПІБ: Василенко Василь Васильович <br> email: teacher@gmail.com <br>
      Телефон: +380506146789
    </p>

  </div>
</div>
</body>
</html>

```

header.html:

```

<div th:fragment="header">
  <nav class="navbar navbar-inverse">
    <div class="container-fluid">
      <ul class="nav navbar-nav">
        <li><a class="navbar-brand" href="/">Електронний курс</a></li>
        <li><a href="/materials">Матеріали</a></li>
        <li><a href="/teacher">Викладач</a></li>
      </ul>
    </div>
  </nav>
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
  <script type="text/javascript" src="js/menuClick.js"></script>
</div>

```

materialItem.html:

```

<div th:fragment="materialItem(panelTitle, panelContent, materialTitle,
materialHref)">
  <div class="panel panel-default">
    <div class="panel-heading">
      <h3 class="panel-title" th:text="{panelTitle}" />
    </div>
    <div class="panel-body">
      <h5>Анотація</h5>
      <p th:text="{panelContent}" />
      <h5>Теоретичний навчальний матеріал</h5>
      <a target="_blank" th:href="{materialHref}" th:text="{materialTitle}" />
    </div>
  </div>
</div>

```

materialItemWithTasks.html:

```

<div th:fragment="materialItemWithTasks(panelTitle, panelContent, materialTitle,
materialHref, moduleTaskHref, moduleTaskTitle)">
  <div class="panel panel-default">
    <div class="panel-heading">
      <h3 class="panel-title" th:text="{panelTitle}" />
    </div>
    <div class="panel-body">
      <h5>Анотація</h5>
      <p th:text="{panelContent}" />
      <h5>Теоретичний навчальний матеріал</h5>
      <a target="_blank" th:href="{materialHref}" th:text="{materialTitle}" />
      <h5>Модульний контроль</h5>
      <a target="_blank" th:href="{moduleTaskHref}" th:text="{
{moduleTaskTitle}" />
    </div>
  </div>
</div>

```

materialItemWithTest.html:

```

<div th:fragment="materialItemWithTest(panelTitle, panelContent, testTitle, testHref,
materialTitle, materialHref)">
  <div class="panel panel-default">
    <div class="panel-heading">
      <h3 class="panel-title" th:text="{panelTitle}" />
    </div>
  </div>

```

```

</div>
<div class="panel-body">
  <h5>Анотація</h5>
  <p th:text="{panelContent}" />
  <h5>Теоретичний навчальний матеріал</h5>
  <a target="_blank" th:href="{materialHref}" th:text="{materialTitle}" />
  <h5>Модульний контроль</h5>
  <a target="_blank" th:href="{testHref}" th:text="{testTitle}" />
</div>
</div>
</div>

```

menuClick.js:

```

$(function () {
  setNavigation();
});

function setNavigation() {
  var path = window.location.pathname;

  $(".nav a").each(function () {
    var href = $(this).attr('href');
    if (path === href) {
      $(this).closest('li').addClass('active');
    }
  });
}

```