КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА
Факультет комп'ютерних наук та кібернетики

Кафедра теорії та технології програмування

Кваліфікаційна робота

на здобуття ступеня бакалавра

за спеціальністю 122 «Комп'ютерні науки»

на тему:

ВЕБ-ДОДАТОК НАВЧАННЯ JAVASCRIPT ДЛЯ ДІТЕЙ

Виконала студентка 4-го курсу
Бабенко Аліна Павлівна _____
(підпис)

Науковий керівник:
Кандидат фізико-математичних наук
Криволап Андрій Володимирович _____
(підпис)

Засвідчую, що в цій роботі немає запозичень з праць інших авторів без відповідних посилань.
Студент _____

Роботу розглянуто й допущено до захисту на засіданні кафедри теорії та технології програмування
«_____»_____ 2023 р,
протокол №_____
Завідувач кафедри
НІКІТЧЕНКО М.С._____
(підпис)

Київ – 2023

РЕФЕРАТ

Обсяг роботи 42 сторінки, 41 ілюстрації, 13 джерел посилань, 1 додаток. ВЕБ-ЗАСТОСУНОК, НАВЧАННЯ JAVASCRIPT, НАВЧАННЯ ДІТЕЙ, РОЗРОБКА ІНТЕРФЕЙСУ, JAVACRIPT, CSS, REACT, YARN

Об’єктом даної роботи є процес навчання дітей основам JavaScript.

Предметом роботи є веб-додаток навчання основам JavaScript для дітей.

Метою роботи є проектування та розробка веб-додатку навчання основам JavaScript для дітей

Методи розроблення: комп’ютерне моделювання, розробка програмного продукту на основі моделі. Інструменти розроблення: інтегроване середовище розробки Visual Studio Code, мова програмування JavaScript, HTML, CSS, React library, Yarn, MUI library.

Результати роботи: проаналізовано переваги та недоліки використання веб-додатків для навчання різних мов програмування. Розроблено програмний застосунок для навчання дітей JavaScript, який дозволяє користувачам отримати базові знання з цієї мови програмування та вирішувати різного ступеню складності завдання.

Розроблений в дипломній роботі веб-додаток може використовуватися дітьми разом з їх батьками для навчання основам JavaScript.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧЕННЯ4

ВСТУП5

РОЗДІЛ 1. ОГЛЯД НАЯВНИХ НА РИНКУ СИСТЕМ8

1.1 JavaRush8

1.2 CodeCademy10

1.3 CodeMonkey11

РОЗДІЛ 2. ОГЛЯД ВИКОРИСТАНИХ ТЕХНОЛОГІЙ14

2.1 Бібліотека React14

2.2 Мова реалізації JS15

2.3 HTML16

2.4 CSS17

2.5 Бібліотека MUI18

2.6 Yarn19

РОЗДІЛ 3. ПРИЗНАЧЕННЯ ТА ЦІЛІ СТВОРЕННЯ ВЕБЗАСТОСУНКУ20

3.1 Призначення веб-застосунку «IT education for children»20

3.2 Цілі створення веб-застосунку «IT education for children»20

3.3 Вимоги до веб-застосунку «IT education for children»21

3.4 Технічні вимоги до веб-застосунку «IT education for children»22

РОЗДІЛ 4. РЕАЛІЗАЦІЯ ВЕБ-ДОДАТКУ23

4.1 Структура додатку23

4.2 Створення проекту24

4.3 Реалізація застосунку24

РОЗДІЛ 5. ІНТЕРФЕЙС КОРИСТУВАЧА31

ВИСНОВКИ40

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ41

ДОДАТКИ42

Додаток А42

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧЕННЯ

HTML – HyperText Markup Language, мова гіпертекстової розмітки;

JS – мова програмування JavaScript;

CSS – Cascading Style Sheets, каскадні таблиці

стилів;

DOM – Document Object Model, об'єктне представлення вмісту HTML-документу та інтерфейс для управління цим об'єктом;

W3C – World Wide Web Consortium, головна міжнародна організація, що розробляє й впроваджує технологічні стандарти;

MUI – Material UI library;

ВСТУП

Оцінка сучасного стану об’єкта розробки. Сучасний стан веб-сайтів навчання мовам програмування для дітей є досить розмаїтим і динамічним. Останні роки спостерігають значний розвиток таких платформ, існують як великі та відомі проекти, так і менш відомі, але інноваційні ідеї. Деякі з них спеціалізуються на конкретних мовах, таких як Scratch, Python або JavaScript, тоді як інші надають можливість вивчати кілька мов програмування.

Актуальність роботи та підстави для її виконання. Комп'ютерна грамотність і навички програмування стають все більш важливими у суспільстві. Сфера програмування швидко розвивається і має великий попит на кваліфікованих фахівців.

Багато футурологів передбачають, що базові знання з програмування стануть такими ж важливими, як уміння читати і писати. Це обумовлено зростанням впливу технологій на різні сфери життя і широким застосуванням комп'ютерів і програмного забезпечення. Автоматизація та розширення ролі технологій в промисловості, бізнесі та інших галузях, зростання штучного інтелекту, Інтернету речей, робототехніки та інших сучасних технологій створює потребу у фахівцях, які розуміють, як ці технології працюють і як їх використовувати.

JavaScript є однією з найпопулярніших мов програмування, яка використовується для розробки веб-сайтів, веб-додатків та інтерактивних елементів. Навчання JavaScript дозволяє дітям створювати власні веб-сторінки, ігри та інші веб-проекти, що може бути захоплюючим та мотивуючим для них. JavaScript працює на більшості платформ, включаючи комп'ютери, планшети та мобільні пристрої. Це дає дітям можливість експериментувати та розробляти свої проекти на різних пристроях і в середовищах. Також треба зазначити, що JavaScript має простий синтаксис та доступну документацію, що допомагає початківцям швидше освоювати основні концепції програмування. Він також має велику спільноту

розробників, яка надає багато навчальних матеріалів та ресурсів для навчання[10-12].

Вивчення JavaScript змалечку допоможе дітям зрозуміти основні принципи програмування та підготує їх до майбутніх можливостей. Вони зможуть легше освоювати складніші мови програмування і розвивати свої навички швидше. Програмування вимагає логічного мислення, а навчання JavaScript надає можливість розвивати цю навичку. Також вивчення JavaScript допомагає розвивати креативність дітей. Вони можуть створювати свої власні ігри, анімації та веб-сайти, використовуючи цю потужну мову програмування. Це дає їм можливість втілювати свої ідеї в життя та стимулює творчий потенціал.

Мета й завдання роботи. Метою кваліфікаційної роботи являється створення веб-додатку навчання JavaScript для дітей. Для досягнення цієї мети поставлені такі завдання.

- Дослідити вимоги до навчання дітей.
- Проаналізувати існуючі веб застосунки, призначені для навчання дітей різним мовам програмування та визначити сильні та слабкі сторони кожного з них.
- Розробити план навчання JS, який відповідатиме всім вимогам здобуття базових знань мови програмування.
- Розробити завдання для тестування та закріплення знань з JS.
- Розробка інтерфейсу та здійснення програмної реалізації веб-додатку.

Об'єкт, методи й засоби розроблення. Об’єктом розроблення веб застосунку є процес навчання дітей основам мови програмування JS та тестування отриманих знань.

Розробка веб застосунку базувалася на кількох принципах. На початку розробки створюється початкова версія застосунку, яка оцінюється. На основі цієї оцінки допрацьовується додаток, з урахуванням всіх зауважень та побажань. Кожен етап розробки успадковує такий механізм, до того моменту поки не з’явиться додаток, який задовольняє всі поставлені вимоги.

Основною мовою при розробці застосунку був JavaScript. Фреймворком, для спрощення процесу розробки, був обраний React у комбінації з бібліотекою компонент-MUI. Для встановлювання, оновлювання та керування версіями пакетів, був використаний менеджер пакетів Yarn. Також для створення та оформлення веб сторінок використовувались HTML та CSS.

В якості інструменту розробки програмного засобу було обрано Visual Studio Code - редактор вихідного коду, розроблений Microsoft для Windows, Linux та macOS, який є безкоштовним та вільно поширюваним.

РОЗДІЛ 1. ОГЛЯД НАЯВНИХ НА РИНКУ СИСТЕМ

Розглянемо веб застосунки, які пов’язані з навчанням різним мовам програмування, та проаналізуємо їх, щоб з’ясувати переваги та недоліки кожного та сформувати вимоги до застосунку, що розробляється в рамках дипломної роботи.

1.1 JavaRush

На рис. 1 зображено вигляд сайту JavaRush [1] – веб-додаток для вивчення мови програмування Java. Зліва наявна панель для навігації сайтом. Вся необхідна інформація згрупована в розділи, цим самим спрощуючи пошук потрібної інформації.

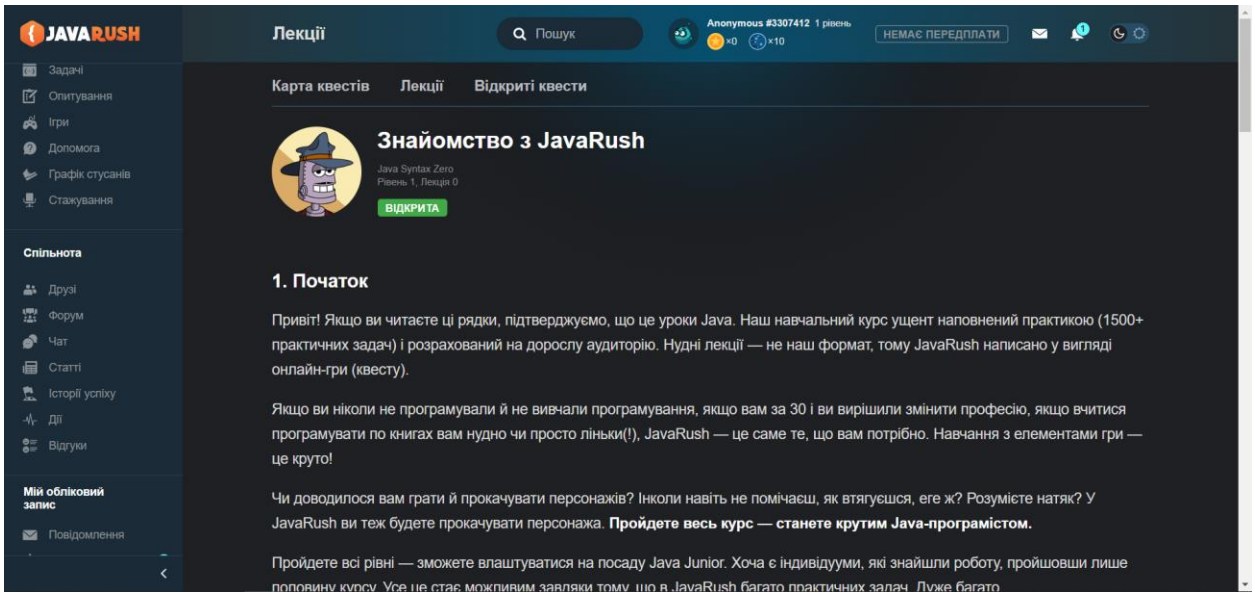

Рисунок 1 – Зовнішній вигляд сайту JavaRush

Даний інтернет-сайт має великий функціонал, багату різноманітність завдань та тестів (див. рис 2). Цей додаток має власну спільноту і можливість стажування. Також із переваг можна виділити наявність особистого кабінету користувача, де зберігається його прогрес. Ще можна виділити наявність можливість вибору мови навчання та дизайн веб-додатку, який адаптований під різні розміри пристроїв.

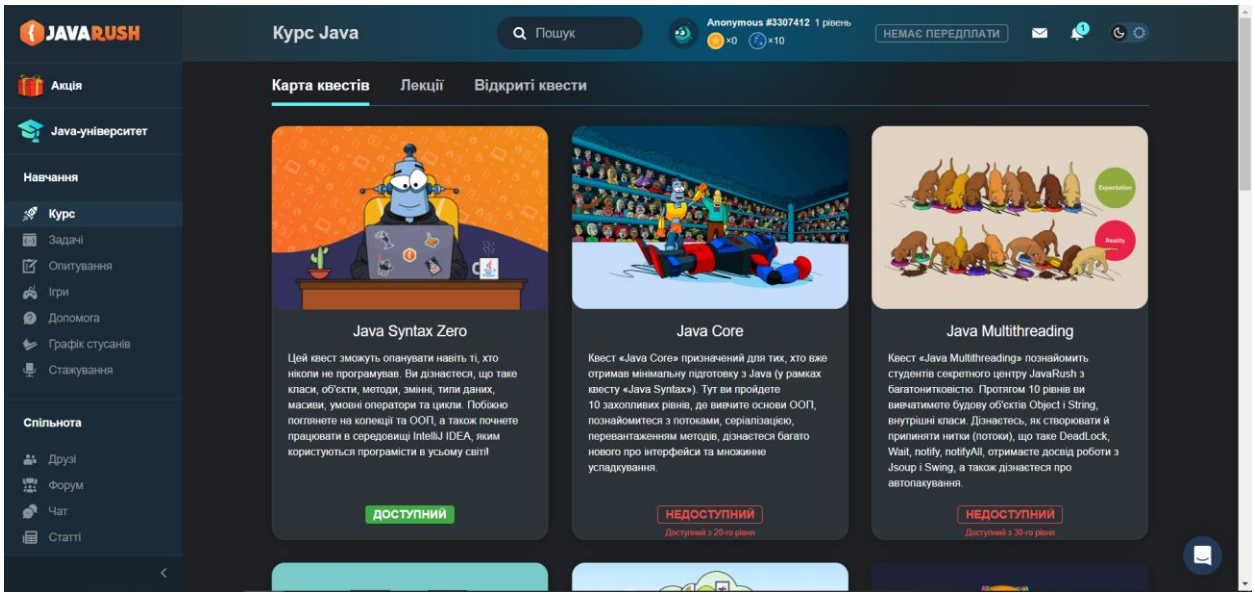

Рисунок 2 – Зовнішній вигляд сторінки курсів

Але цей веб-додаток має недолік у вигляді платних підписок, тому що без них більша частина контенту та функцій є недоступними. І через це у користувача без підписки є великі обмеження у навчанні (див. рис 3).

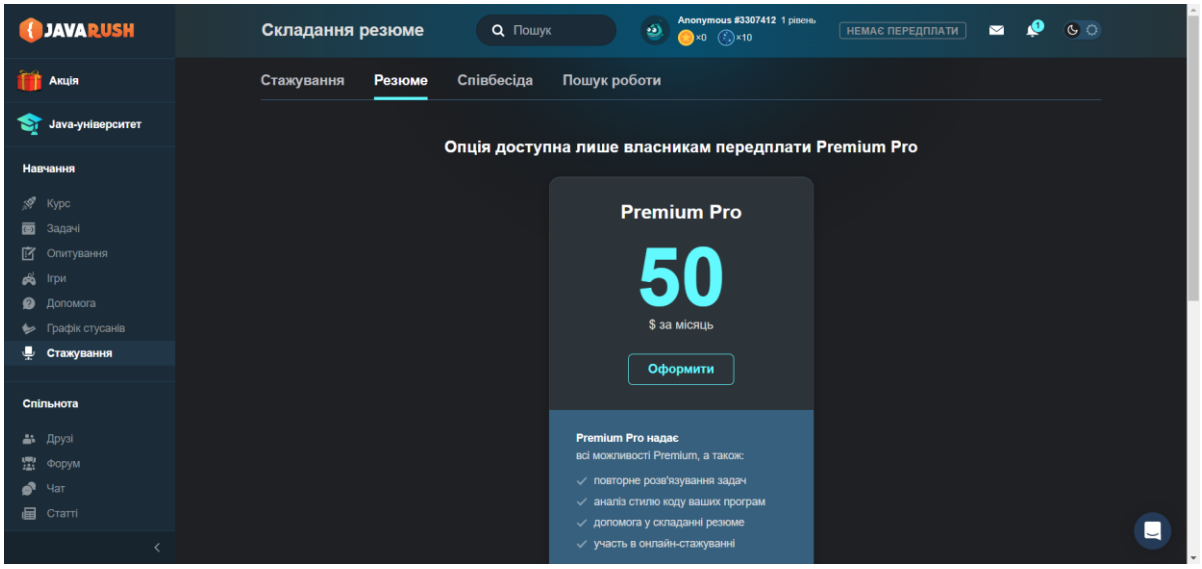

Рисунок 3 – Платна підписка

1.2 CodeCademy

На рис. 4 зображено вигляд головної сторінки CodeCademy[2]-платформа навчання різним мовам програмування. Посеред сторінки є форма для реєстрації особистого кабінету користувача. А зверху є панель навігації завдяки якої є можливість швидко перейти до різних розділів сайту.

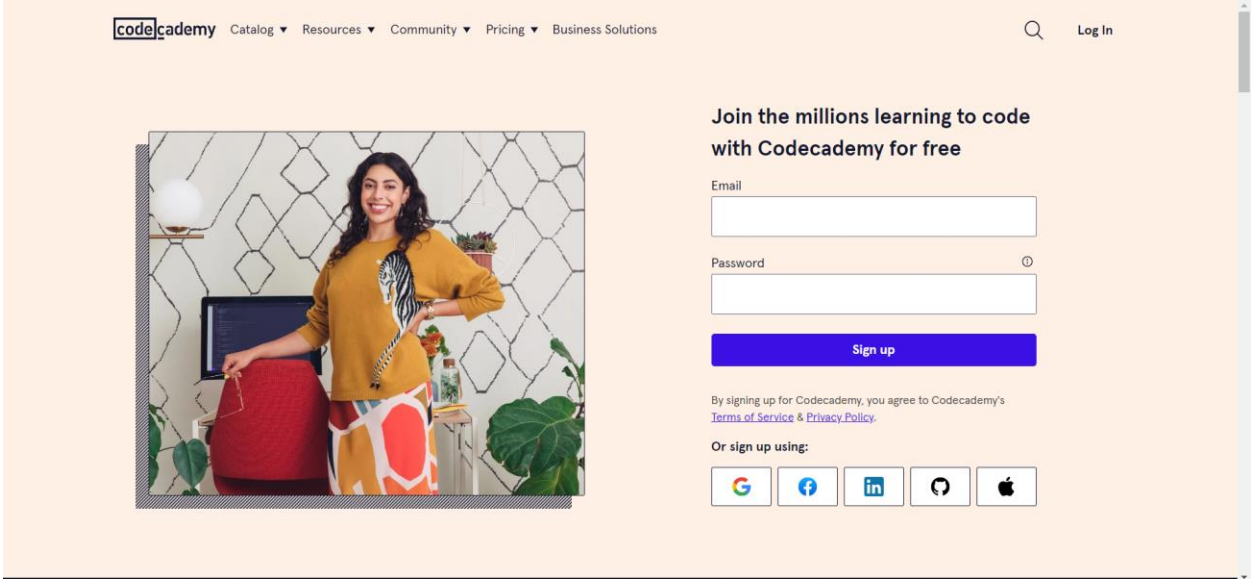

Рисунок 4 – Головна сторінка CodeCademy

Цей веб-застосунок має курси за різними мовами програмування та за різними рівнями кваліфікації(див. рис 5). Також є різні курси для вивчення окремих напрямів у програмуванні, наприклад Data Science або Machine Learning. Також з переваг цього сайту можна виділити мінімалістичний дизайн, який робить процес навчання ще більш комфортнішим.

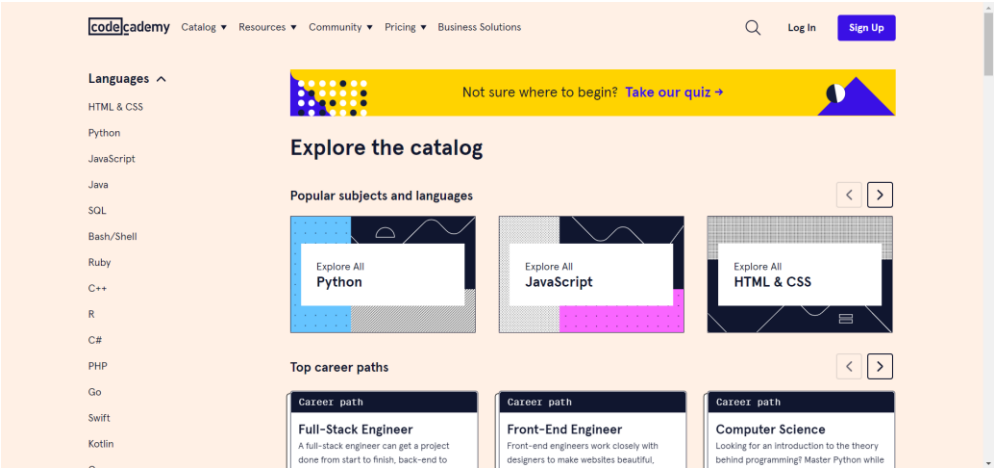

Рисунок 5 – сторінка каталогу курсів

Цей платформа, яка має великий обсяг корисної інформації, важливих знань та власну спільноту, є дуже хорошим сайтом для вивчання різних напрямів та мов програмування. Але через головний недолік - наявність лише одніє мови, кількість можливість користувачів значно зменшується. Також через спосіб подачі інформації та оформлення сайту, він зовсім не є адаптованим для дітей.

1.3 CodeMonkey

На рис. 6 зображено вигляд головної сторінки CodeMonkey[13]- платформа навчання програмування для дітей. Зверху є панель навігації завдяки якої є можливість швидко перейти до різних розділів сайту. Одразу хочеться зазначити дитячий дизайн разом з різними анімаціями та тваринками.

Рисунок 6 – Головна сторінка CodeMonkey

Це освітнє середовище комп'ютерного кодування, яке дозволяє новачкам вивчати концепції та мови комп'ютерного програмування за допомогою курсів-ігор. CodeMonkey призначений для учнів віком від 6 до 14 років (див. рис 7).

Рисунок 7 – Курси CodeMonkey

Більшість курсів націлені на вивчання концепції програмування та розвинення логіки для розуміння кодування (див. рис 8).Також є один курс для вивчення основ Python (див. рис 9).

Рисунок 8 – Курс на CodeMonkey

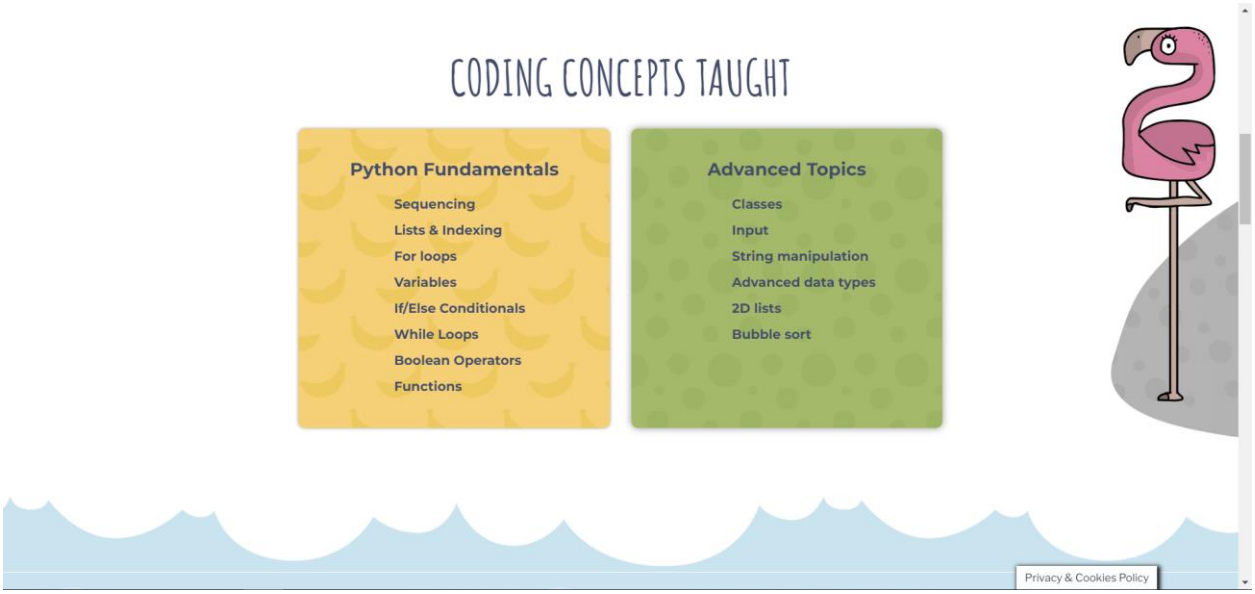

Рисунок 9 – Курс Python на CodeMonkey

Проаналізувавши існуючі інтернет-сайти, навчання різних мов програмування, можна зробити висновок, що всі вони відрізняються роботою дизайнера та функціоналом, можливістю вивчати як один так і декілька мов. Деякі веб застосунки націлені тільки на дорослих користувачів, а деякі з них адаптовані для дітей.

Було прийнято рішення для реалізації власного застосунку, який буде націлений на дітей, і буде спеціалізуватися на навчанні однієї мови програмування - JavaScript.

РОЗДІЛ 2. ОГЛЯД ВИКОРИСТАНИХ ТЕХНОЛОГІЙ

Для розробки та реалізації веб-додатку навчання JS для дітей, використано технології:

- Бібліотека React;
- Мова реалізації JS;
- HTML;
- CSS;
- бібліотека MUI;
- Yarn;

2.1 Бібліотека React

Для створення веб-застосунку було обрано бібліотеку React[3]. Це відкрита JavaScript бібліотека для розробки інтерфейсів користувача. Вона дозволяє створювати компоненти, які відображаються на веб-сторінці і автоматично оновлюються при зміні даних. React був розроблений Facebook і перші версії вийшли в 2013 році. З тих пір він став однією з найпопулярніших бібліотек для розробки веб-додатків. Його популярність гарантує наявність великої та активної спільноти розробників, документації, онлайн-ресурсів і багатьох сторонніх бібліотек і розширень.

Основною ідеєю React є поділ інтерфейсу на незалежні компоненти, які можуть бути перевикористані і збудовані в складніші структури. Кожен компонент має внутрішній стан і може приймати властивості (props) від батьківського компонента. Зміни в стані компонента призводять до перерендерингу (оновлення) інтерфейсу.

React використовує віртуальний DOM (Document Object Model), який представляє структуру інтерфейсу в пам'яті. Коли дані змінюються, React порівнює старий віртуальний DOM з новим і визначає набір мінімальних змін, які потрібно внести в реальний DOM, щоб оновити інтерфейс. Цей підхід

дозволяє забезпечити ефективну роботу з інтерфейсом, навіть у великих додатках зі складними структурами.

Одним із ключових понять у React є "однієї напрямок потік даних" (one-way data flow). Дані передаються з батьківського компонента до дочірніх через властивості (props). Дочірні компоненти не можуть безпосередньо змінювати дані, вони можуть тільки повідомляти батьківський компонент про зміни через спеціальні функції зворотного виклику (callback functions).

React також надає можливість використовувати JSX - спеціальний синтаксис, який поєднує JavaScript і HTML. JSX дозволяє описувати структуру інтерфейсу прямо в коді JavaScript, що полегшує розробку та зроблює код більш зрозумілим.

React є гнучкою бібліотекою, яка дозволяє легко розширювати її функціональність за допомогою сторонніх ресурсів. Існують багато розширень, які додають нові функції, такі як маршрутизація, керування станом або анімація, що полегшує розробку складних додатків. Наприклад, React Router дозволяє створювати роутинг в React-додатках, а Redux надає засоби для управління станом додатка.

React є дуже популярним інструментом для розробки сучасних веб-додатків. Він має велику та активну спільноту розробників, що забезпечує доступ до багатьох ресурсів, документації та відкритих вихідних кодів проектів. React використовується в численних компаніях і веб-продуктах, що підтверджує його ефективність та надійність.

Саме тому для розробки клієнтської частини веб-додатка було обрано використовувати саме цю бібліотеку.

2.2 Мова реалізації JS

В якості мови реалізації веб-застосунку було обрано JS. JavaScript є однією з найпопулярніших та широко використовуваних мов програмування. Вперше випущений в 1995 році компанією Netscape, JS спочатку задумувався

як мова для програмування на стороні клієнта веб-сторінок, але з часом його застосування розширилося на багато інших платформ. JS можна використовувати як на клієнтській стороні (у веб-браузерах), так і на серверній стороні (за допомогою Node.js). Це робить його універсальним і дає можливість розробляти повноцінні веб-додатки на обох сторонах. Через це JS має велику та активну спільноту розробників. Можна знайти багато ресурсів, документації, блогів і форумів, де можна отримати підтримку, поради та відповіді будь-які запитання.

При розробці інтерфейсів JS дозволяє додавати інтерактивність до веб-сторінок, дозволяючи користувачам взаємодіяти з елементами сторінки. За його допомогою можна обробляти події, валідувати дані, змінювати вміст сторінки та багато іншого. JS дає доступ до багатьох браузерних API, які дозволяють взаємодіяти з елементами сторінки, маніпулювати DOM, виконувати асинхронні запити до сервера та багато іншого. Це дає можливість створювати багатофункціональні веб-додатки безпосередньо в браузері.

JS має велику кількість бібліотек і фреймворків, які полегшують розробку веб-додатків. Наприклад, React - вибраний фреймворк для розробки фронтенду у кваліфікаційній роботі. За допомогою нього можна створювати складні та потужні веб-додатки з ефективним управлінням станом, маршрутизацією, керуванням станом та багатьма іншими функціями.

JS є потужним і розширюваним інструментом для розробки веб-додатків, і його використання в дипломній роботі надав широкий набір можливостей для створення інтерактивних та ефективних додатків.

2.3 HTML

HTML (HyperText Markup Language) є основною мовою розмітки для створення веб-сторінок. Вона використовується для створення структури і вигляду веб-сторінок та надання вмісту для відображення в браузерах.

HTML дозволяє визначити структуру веб-сторінки за допомогою різних елементів, таких як заголовки, параграфи, списки, таблиці і т.д. Ці елементи допомагають визначити логічну структуру вмісту сторінки. Ця мова використовує теги для огортання і визначення різних частин вмісту. Наприклад, `<h1>` використовується для визначення найбільшого заголовка, `<p>` - для відображення абзацу тексту. Кожен тег може також мати атрибути, які надають додаткові властивості елементу. А за допомогою тега `<a>` можна створювати гіперпосилання, які дозволяють користувачам переходити до інших сторінок або ресурсів. Також можна використовувати атрибут `href` для вказівки URL-адреси, до якої буде здійснено перехід.

HTML надає можливість створювати форми, які дозволяють користувачам вводити дані. Можно використовувати тег `<form>` для огортання форми і різні типи введення, такі як текстове поле `<input type="text">`, чекбокси `<input type="checkbox">`, радіокнопки `<input type="radio">` і багато інших.

Мова HTML є основою для будь-якої веб-сторінки і є важливим елементом веб-розробки. Ці аспекти HTML були використаними при розробці веб-сторінок у кваліфікаційній роботі.

2.4 CSS

CSS (Cascading Style Sheets) є мовою опису стилів, яка використовується для задання зовнішнього вигляду веб-сторінок, створених з використанням HTML. Ця мова грає важливу роль у створенні привабливого та естетичного вигляду сторінок. CSS має ключові аспекти, які використовуються при розробці.

CSS використовує селектори для вибору елементів на веб-сторінці, до яких будуть застосовуватися стилі. Селектори можуть бути базовими (наприклад, назва елемента, клас або ідентифікатор) або комбінованими (наприклад, вибір елементів в певному контексті або на підставі їх взаємного

відношення). Також CSS містить широкий набір властивостей, які можна застосовувати до веб-елементів, наприклад, кольори, шрифти, розміри, відступи, рамки, фон і т.д. Кожна властивість має своє значення, яке визначає, як буде виглядати веб-елемент. CSS використовує принцип каскаду, де стилі можуть бути задані на різних рівнях ієрархії, і вони можуть спадати на дочірні елементи. Це означає, що є можливість задавати загальні стилі для батьківських елементів, які будуть автоматично застосовуватися до їх дочірніх елементів. CSS дозволяє використовувати медіа-запити для визначення стилів, які будуть застосовуватися залежно від характеристик пристрою або екрана, на якому відображається веб-сторінка. Це дозволяє створювати адаптивні дизайни, які оптимізуються для різних пристроїв і розмірів екранів.

Також CSS надає можливість створювати анімації та переходи для елементів на веб-сторінці. Можно задавати ключові кадри, швидкість, режими анімації та багато іншого, щоб створити рухливі та привабливі ефекти.

2.5 Бібліотека MUI

Material-UI[4] є популярною бібліотекою компонентів для React, яка може спростити розробку користувацького інтерфейсу, забезпечити сучасний та привабливий дизайн і полегшити роботу з React-компонентами. Ця бібліотека базується на дизайні "Material Design" від Google, який пропонує чітку і консистентну систему дизайну. Це дозволяє створювати сучасні та естетичні інтерфейси, які відповідають сучасним стандартам дизайну. Також Material-UI має широку та активну спільноту розробників. Це означає, що можна легко знайти документацію, приклади коду, плагіни та розширення, а також отримувати підтримку від інших розробників.

Ось деякі переваги Material-UI перед іншими бібліотеками:

1. Готові компоненти: Material-UI надає широкий набір готових компонентів, таких як кнопки, форми, таблиці, спливаючі вікна, меню, картки та багато інших. Це дозволяє розробникам швидко і легко

будувати стильні та функціональні інтерфейси, не витрачаючи багато часу на розробку компонентів з нуля.

2. Розширені можливості налаштування: Material-UI дозволяє налаштовувати компоненти залежно від потреб. Є можливість змінювати кольори, типографію, розміри та інші параметри за допомогою тем і налаштувань. Це дозволяє легко і швидко впроваджувати власний бренд або стиль у будь-якому додатку.
3. Реактивність: Material-UI розроблено для використання з React, що дозволяє створювати динамічні та реактивні інтерфейси. Розробники мають можливість легко оновлювати стан компонентів, використовувати анімацію та переходи, та взаємодіяти з іншими бібліотеками та фреймворками React.

2.6 Yarn

Yarn[5] - це менеджер залежностей для проектів з використанням JavaScript. Він призначений для керування бібліотеками, пакетами та їх версіями, які використовуються у вашому проекті. Yarn був розроблений Facebook з метою заміни npm (Node Package Manager) як інструменту для керування залежностями.

Основна перевага Yarn полягає у його швидкості та надійності завантаження пакетів. Він використовує кешування та оптимізовані алгоритми для покращення продуктивності завантаження. Крім того, Yarn має можливість паралельного завантаження пакетів, що дозволяє прискорити процес інсталяції.

Yarn має інтуїтивний і простий у використанні інтерфейс командного рядка, де можно виконувати команди, такі як **yarn install** для встановлення всіх залежностей з файлу **package.json** або **yarn add** для додавання нового пакету до проекту. Він також забезпечує можливість оновлювати та видаляти пакети.

РОЗДІЛ 3. ПРИЗНАЧЕННЯ ТА ЦІЛІ СТВОРЕННЯ ВЕБЗАСТОСУНКУ

3.1 Призначення веб-застосунку «IT education for children»

Призначенням веб-застосунку **«IT education for children»** є використання проекту для навчання дітей основам мови програмування JS, викладення теоретичної бази з JS, створення зручного довідника з основною інформацією, тестування здобутих знать на практиці та створення гри, у процесі якої діти будуть закріпляти всі отримані навички.

Кваліфікаційна бакалаврська робота передбачає:

- дослідження методів та методик процесу навчання дітей;
- створення спеціалізованого плану и завдань для вивчення JS;
- аналіз методів, методик, технологій і моделей, що використовуються для створення веб-застосунів;
- аналіз існуючих програмних засобів;
- проектування та програмну реалізацію застосунку.

3.2 Цілі створення веб-застосунку «IT education for children»

Вебзастосунок **«IT education for children»** створюється з метою:

- ознайомлення дітей з програмуванням;
- надання дітям базових знань з JS;
- закріплення отриманих теоретичних знань завдяки практичним вправам;
- ствjрення гри для зацікавлення дітей у процессі отримання нових знань;

- забезпечення зручного інтерфейсу та викладення інформації для дітей.

3.3 Вимоги до веб-застосунку «IT education for children»

Застосунок повинен бути розроблений враховувати специфіку та потреби цільової аудиторії – дітей [6-8].

Важлива роль у цьому займає дизайн. Використання яскравих та насичених кольорів привертає увагу дітей і робить сайт привабливим. Основні та додаткові кольори – червоний, синій, жовтий, зелений, фіолетовий і помаранчевий – усі це «щасливі» кольори, і популярні варіанти та колірні схеми можуть включати їх усі та багато іншого. Потрібно обмежити використання пастельних і дорогоцінних тонів або використовувати їх в поєднанні з насиченими кольорами.

Діти хочуть, щоб веб-сайт був щасливим, тому загальний настрій веб-сайту має бути таким, як і яскраві та «щасливі» кольори. Усміхнені обличчя, життєрадісні характери, енергійні жести та позитивні слова – є гарними елементами для дитячого сайту. Використання впізнаваних форм подобається дітям, оскільки це заохочує та зміцнює процес навчання. Тварини, форми з природи та предмети з дому знайомі та впізнавані дітям, оскільки вони складають більшу частину їхнього візуального та тактильного світу. Також додавання помічника на дитячих сайтах для навчання є важливим засобом для забезпечення більшої інтерактивності та залучення дітей до навчального процесу. Він може виступати в ролі віртуального вчителя, наставника чи просто дружнього персонажа, який сприяє активному взаємодії з дітьми.

Також потрібно залучити анімацію, яка відіграє важливу роль у покращенні користувацького досвіду та ефективності навчання. Анімація привертає увагу дітей і створює захопливу атмосферу. Рухомі елементи, переходи та інтерактивність роблять сайт цікавішим і спонукають дітей до активності. Складні концепції можна легше зрозуміти та запам'ятати завдяки анімації. Вона може допомогти візуалізувати процеси, динамічність та взаємозв'язки між елементами, що полегшує сприйняття матеріалу. Анімація

може бути використана для створення інтерактивних вправ, ігор та завдань, які допомагають дітям навчатися через діяльність. Рухомі елементи можуть слугувати підказками, поясненнями або нагородами за правильні відповіді. Також анімація дозволяє створити емоційне зв'язування з дітьми. Анімовані персонажі, які виконують різні дії або виражають емоції, можуть бути цікавими та привабливими для дітей. Вони можуть спонукати до емоційного залучення та відчуття співпереживання.

Основною метою інтерактивних функцій є навчання та розвага дітей. Використання відео, ігор та занять може покращити їх навички аудіювання, читання та арифметики. Роздруковані діяння, такі як розфарбовування аркушів, сприяють розвитку творчих здібностей, координації та уяви. Молодь показує інтерес до навчання, а веб-сайт надає чудову можливість зробити цей процес веселим та привабливим. Ігри, завдання та головоломки можуть бути використані для навчання та тренування, а також можна встановлювати нагороди, наприклад, значки та рівні, щоб заохотити до досягнень.

Батьки іноді відчувають певні обтяження від веб-сайтів, тому важливо переконати їх у тому, що конкретний веб-сайт є безпечним, надійним і відповідає віковим потребам. Для цього необхідно створити розділ для батьків, де буде описана мета веб-сайту та надана необхідна інформація, яка може бути корисною для батьків.

3.4 Технічні вимоги до веб-застосунку «IT education for children»

Розробка проекту повинна орієнтуватися на браузери Google Chrome, Firefox, Edge і Opera. Таким чином, щоб в цих браузерах дизайн відображався однаково, і вся функціональність працювала коректно. Тобто веб додаток повинен відповідати стандарту W3C [9]. У цьому стандарті містяться тільки ті інструменти, які підтримуються всіма браузерам (більшістю).

Операційна система: Windows, Linux, macOS.

Застосунок повинен бути розроблений використовуючи технологію React та використовуючи мови JS, HTML, CSS**.**

РОЗДІЛ 4. РЕАЛІЗАЦІЯ ВЕБ-ДОДАТКУ

4.1 Структура додатку

Веб-додаток навчання JavaScript для дітей «IT education for children» повинен мати дизайн із використанням кольорової палітри (див. додаток А), допоміжника котика-програміста та структуру:

1. Розділ «Trip», в якому буде знаходитись основна частина теорії з мови програмування JS. Вся інформація потрібна бути поділена на теми і мати чітку структуру. Знання потрібні викладатися послідовно саме і тому вся база знань буде представлена у вигляді «пригоди», тобто після проходження однієї теми буде відкриватися доступ до нової, більш складної.
2. Розділ «Zoo» - це розділ-гра. На цієї сторінці будуть зображені тварини у зоопарку, і деякі з них будуть мати сумний настрій. Мета користувача-зробити усіх тварин щасливішими за допомогою проходження завдань від окремих тваринок: це будуть як і тести по JS для закріплення отриманих теоретичних знань, так і грайливих завдань для підняття енергійності та зацікавленості дитини.
3. Розділ «Tasks», який буде на сторінці відображати по одному практичні завдання з JS, щоб користувач зміг перевірити свої знання. На цій сторінці потрібно мати також можливість отримати підказку по вирішенню тесту.
4. Розділ «Handbook», який повинен мати коротко надані всі базові знання з JS. Вся інформація повинна бути поділена на теми, щоб користувач міг швидко знайти потрібне.
5. Розділ «For parents»-сторінка, де буде зазначена вся необхідна інформація для батьків дитини, де буде вказані цілі та методики навчання цього застосунку.
6. Домашня сторінка, на якій буде коротка інформація про JS та програмування, та буде спонукати користувача почати навчання.

4.2 Створення проекту

Розробку застосунку почнемо із створення нового React проекту . Для цього ми повинні спочатку встановити Node версії >= 8.10 та npm версії >= 5.6 на вашому комп'ютері. Для створення проекту виконайте (див. рис 10):

```
npx create-react-app my-app
cd my-app
npm start
```

Рисунок 10 – Create app

Після цього ми маємо новий React проект, відкритий на localhost:3000. (див. рис 11,12)

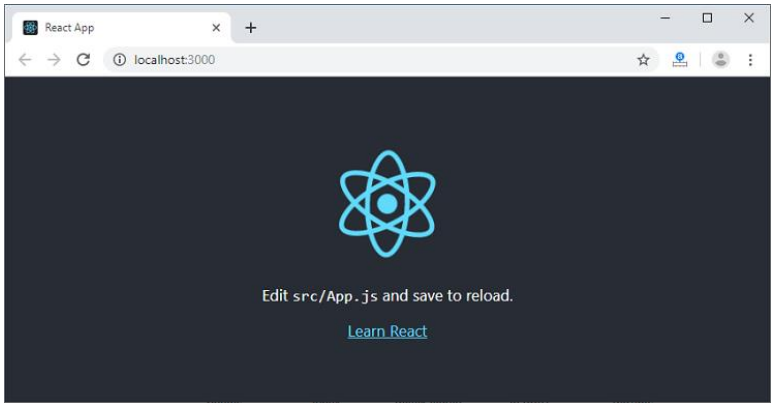

Рисунок 11– React app

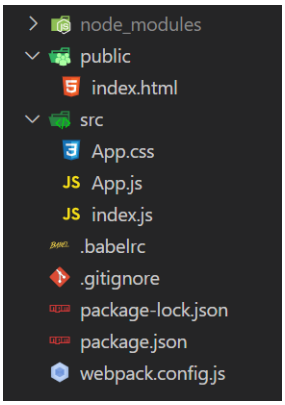

Рисунок 12 – Files structure

Після цього можемо встановити yarn для більш зручної роботи з проектом (див. рис.13,14).

```
npm install --global yarn
```

Рисунок 13 – Install yarn

```
yarn start
```

Рисунок 14– Yarn start

Тепер можемо починати розробку інтерфейсу нашого веб-застосунку за допомогою JavaScript, HTML, CSS.

4.3 Реалізація застосунку

На рис. 15 файлова структура проекту. У папці app знаходяться всі веб сторінки та компоненти, поділені за відповідними розділами, а в папці ui знаходяться ui-компоненти, які використовуються у різних компонентах та основні стилі проекту. Наприклад компоненти Text, Button, тощо (див. рис 16).

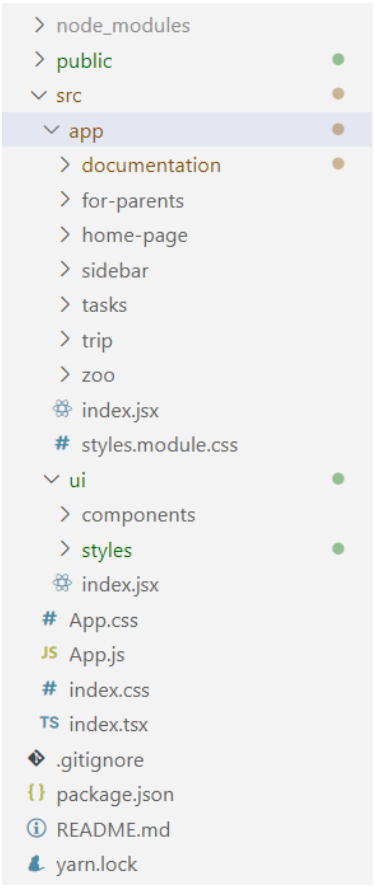

Рисунок 15 – Структура проекту

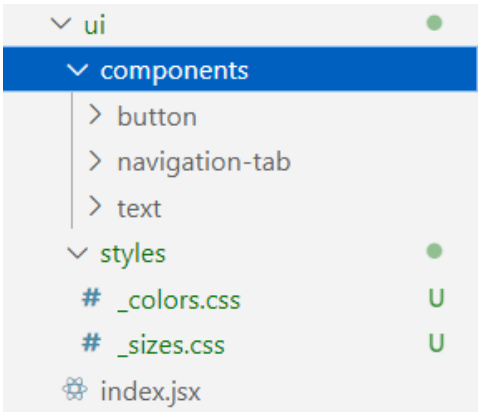

Рисунок 16 – Папка UI

На рис. 17 папка documentation. У цій папці є файл index на якому знаходиться відображення всієї сторінки. Також є папка JSdata де за розділами інформації, створені окремі файли.

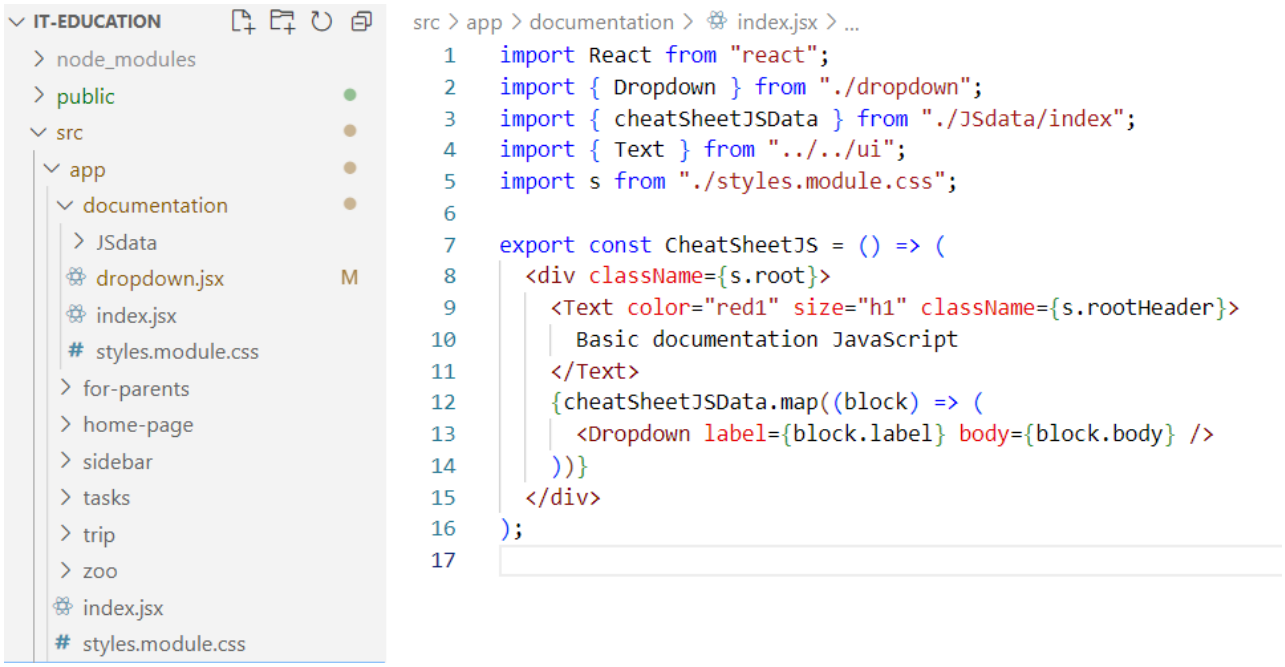

Рисунок 17 – Папка documentation

Кожний розділ відображається у вигляді згортаємого блоку-компонента Dropdown (див. рис 18).

src > app > documentation > dropdown.jsx > Dropdown

```jsx
import React, { useState } from "react";
import Collapse from "@mui/material/Collapse";
import KeyboardArrowDownOutlinedIcon from "@mui/icons-material/KeyboardArrowDownOutlined";
import KeyboardArrowUpOutlinedIcon from "@mui/icons-material/KeyboardArrowUpOutlined";
import { Text } from "../../ui";
import s from "./styles.module.css";

export const Dropdown = ({ label, body }) => {
  const [isOpen, setOpen] = useState(false);
  return (
    <div className={s.dropdown}>
      <button onClick={() => setOpen(!isOpen)} className={s.dropdownHeader}>
        <Text size="l" color="black">
          {label}
        </Text>

        {isOpen ? (
          <KeyboardArrowUpOutlinedIcon />
        ) : (
          <KeyboardArrowDownOutlinedIcon />
        )}
      </button>

      <Collapse in={isOpen} className={s.dropdownBody}>
        {body}
      </Collapse>
    </div>
  );
};

```

Рисунок 18 – Компонента Dropdown

В папках for-parents та home-page знаходяться відповідно ці сторінки разом із файлами зі стилями (див. рис 19).

src > app > home-page > index.jsx > ...

```jsx
import React from "react";
import { Link } from "react-router-dom";
import { Text, Button } from "../../ui";
import s from "./styles.module.css";

export const HomePage = () => (
  <div className={s.root}>
    <div className={s.header}>
      <Text color="black" size="h1" className={s.headerText}>
        Hi dear friend! My name is Kitty.
      </Text>
      <img src="cat.svg" alt="cat" />
    </div>

    <div className={s.block}>
      <img src="laptop.svg" alt="laptop" />
      <Text color="black" size="l">
        The modern world is built on technology. And it is thanks to programm
        that it has become so. All games, applications, sites and the Interne
        itself were created by programmers. And it all looks like magic!
      </Text>
    </div>

    <div className={s.blockJS}>
      <Text color="orange1" size="l">
        I'm clever cat-programmer and i want to share my knowleges with you!
        suggest starting with  
      </Text>
      <Text color="black" size="l" className={s.js}>
        JavaScript
      </Text>
    </div>
    <Text color="black" size="l" className={s.blockJS}>
      JS is one of the most popular programming languages. With it, you can d
      anything. And I want to teach you the basics of this language. If you a
      interested then follow me!
    </Text>
    <Link to="/trip" style={{ margin: "auto" }}>
```

src > app > for-parents > index.jsx > ForParents

```jsx
import React from "react";
import { Text } from "../../ui";
import s from "./styles.module.css";

export const ForParents = () => (
  <div className={s.root}>
    <img src="family.svg" alt="" width={250} />
    <Text size="h1" color="green1">
      Welcome, Parents!
    </Text>
    <Text size="m">
      We are excited to have you here on our site dedicated to teaching your
      children the programming language, Javascript. In this digital age, it
      becoming increasingly important for children to learn coding skills tha
      will help them in their future careers and everyday life.
      <br />
      Javascript is a popular and versatile programming language that is used
      create dynamic websites, web applications, and games. With Javascript,
      your child can create fun and interactive projects, develop
      problem-solving skills, and enhance their logical thinking abilities.
      <br />
      Our program is designed to make learning Javascript easy and engaging f
      children. We use visual and interactive tools that allow children to le
      at their own pace and have fun while doing it.
      <br />
      By learning Javascript, your child will develop skills that will benefi
      them not only in the field of computer science but in many other areas
      well. Coding promotes critical thinking, problem-solving, and creativit
      all of which are valuable skills that will serve them well in their
      academic and professional lives.
    </Text>
    <Text size="m" color="red1">
      We believe that every child has the potential to learn to code and unle
      their creativity. We are here to support you and your child every step
      the way on their coding journey. Thank you for choosing our program, an
      we look forward to working with you and your child.
    </Text>
  </div>
```

Рисунок 19 – Домашня сторінка та сторінка для батьків.

На рис. 20 показана реалізація навігаційної частини проекту – компонента Sidebar.

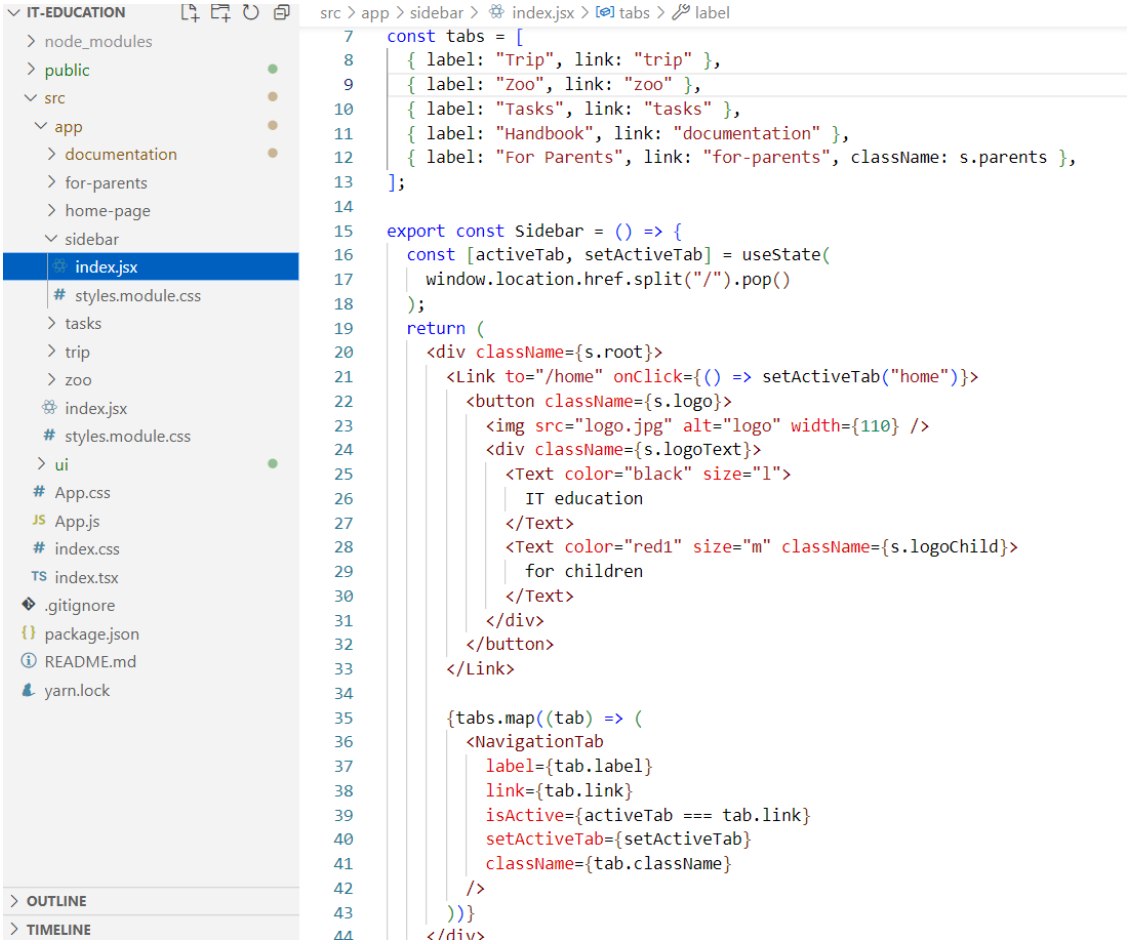

Рисунок 20 – Sidebar

На рис. 21 папка Tasks. У цій папці є файл index на якому знаходиться відображення всієї сторінки. Завдання відображаються по одному у рандомній послідовності, тобто кожного разу береться рандомне завдання із массива tasksData. Також є папка data де для кожного завдання є свій окремий файл.

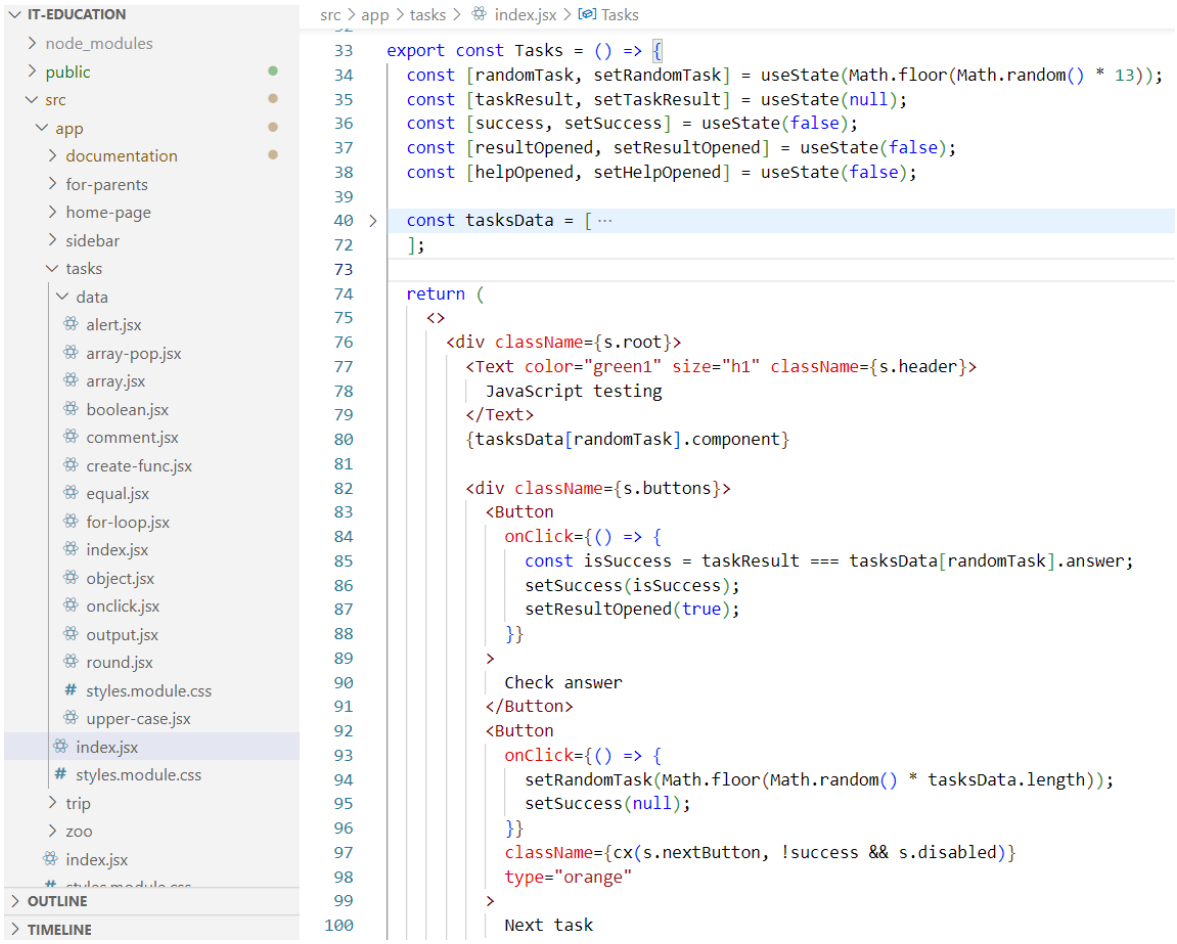

Рисунок 21 – Tasks

На рис. 22 папка Trip. У цій папці є файл index на якому знаходиться відображення всієї сторінки. Є папка steps, де по окремих файлах знаходиться реалізація окремих розділів.

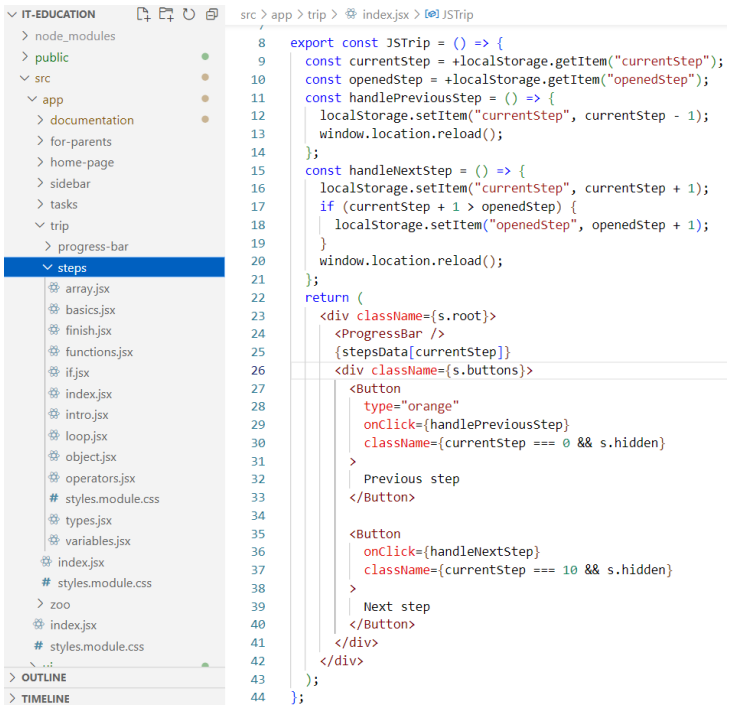

Рисунок 22 – Папка trip

Також папка progress-bar у якому реалізована компонента, яка відповідає за відображення прогресу користувача (див. рис 23).

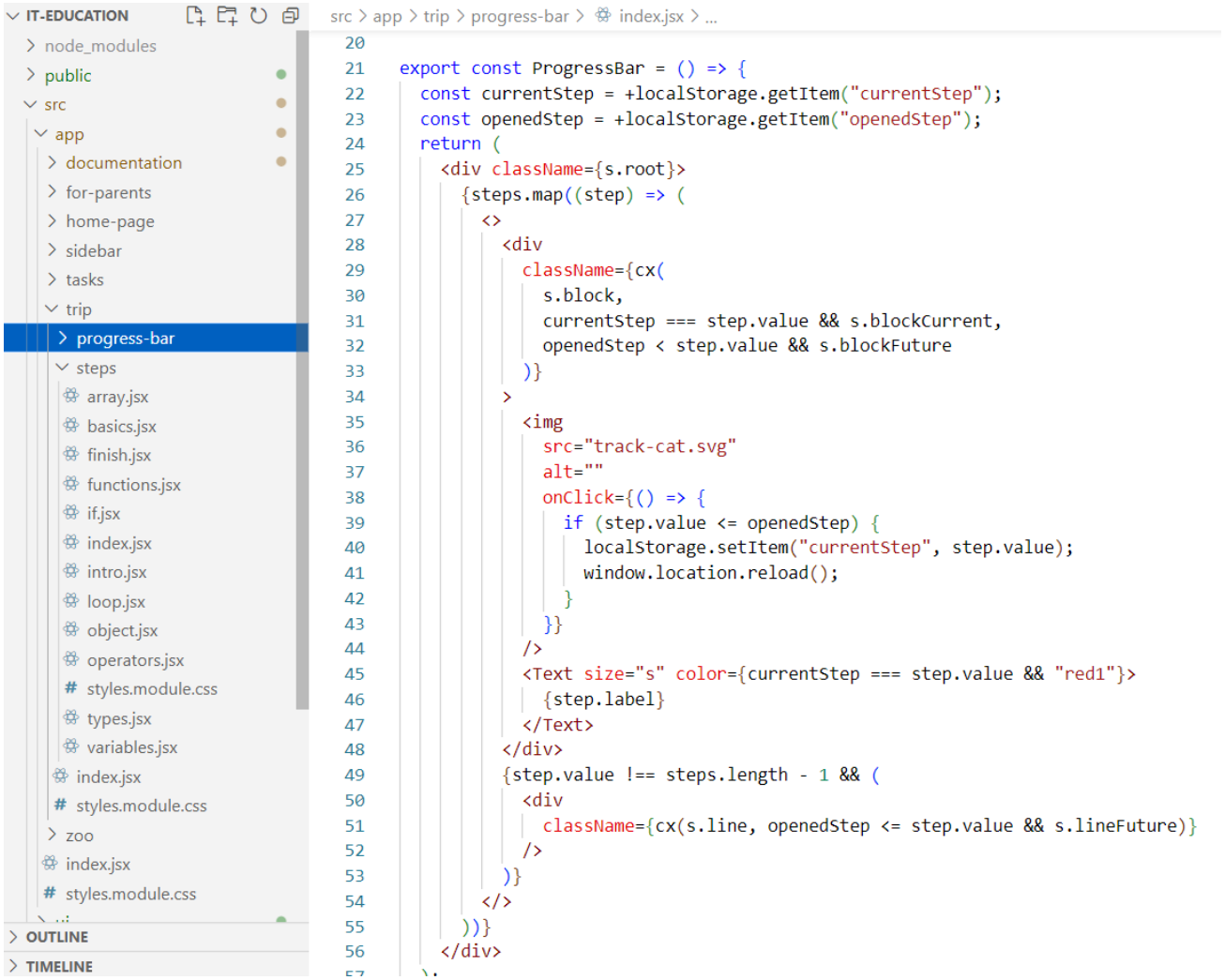

Рисунок 23 – ProgressBar

На рис. 24 папка Zoo. У цій папці є файл index на якому знаходиться відображення всієї сторінки-гри. Є папка animals, де по окремих файлах знаходиться реалізація окремих тварин разом із їх завданнями.

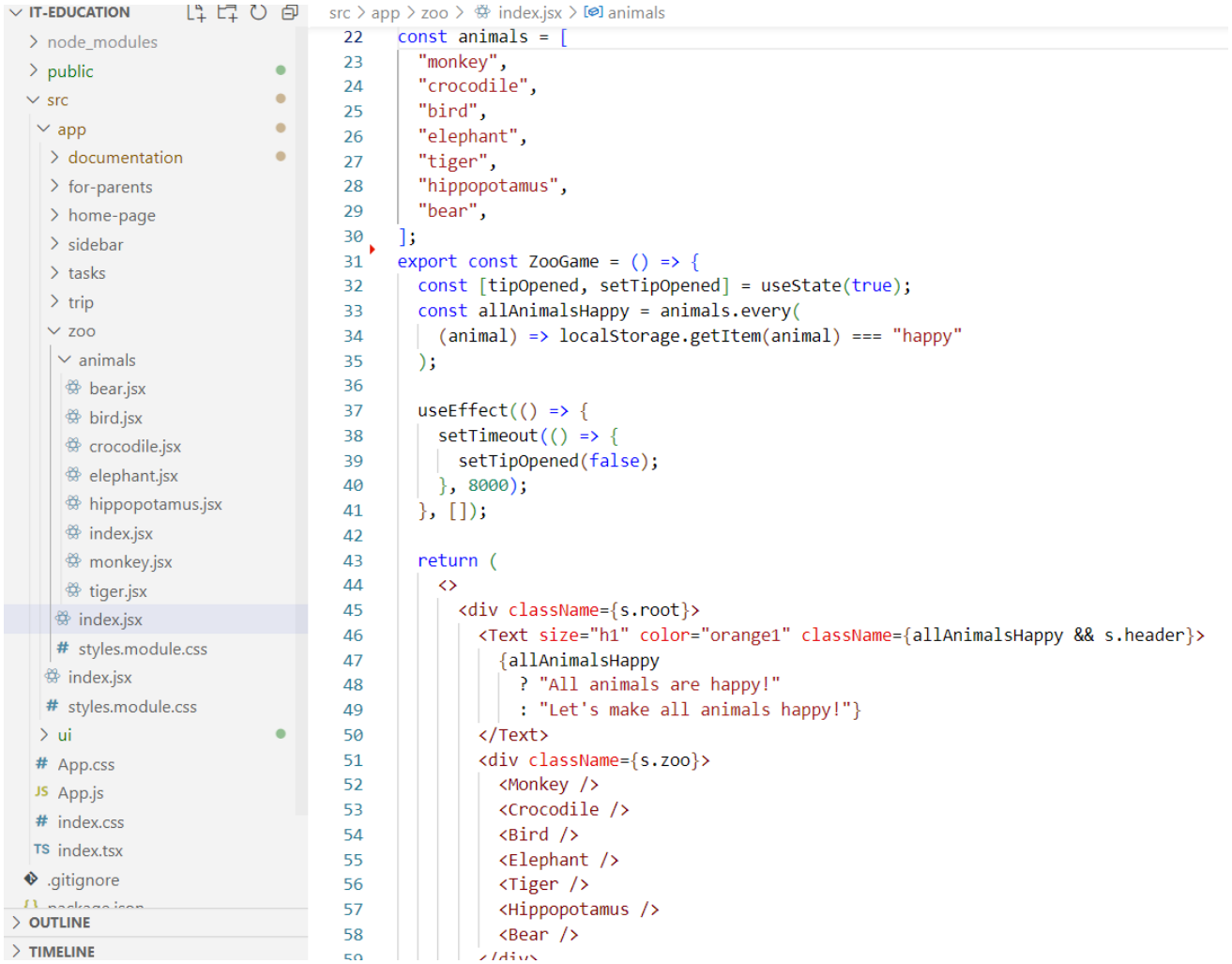

Рисунок 24 – Zoo

РОЗДІЛ 5. ІНТЕРФЕЙС КОРИСТУВАЧА

Веб-додаток відкривається з домашньої сторінки (див. рис.25). На цій сторінці відображення короткої інформації про програмування та котик-програміст, який спонукає на вивчення програмування, і буде допомогати у процесі навчанні. Також є анімована кнопка, яка переадресовує на розділ “Trip”.

Також зліва є навігаційна панель, завдяки якої можна шкидко перейти на потрібний розділ.

Рисунок 25 – Home сторінка

Переходимо на розділ “Trip” (див. рис.26). Цей розділ складається з різних етапів, які поступово відкриваються протягом їх проходження. Зверху бачимо progress-bar, який наглядно демонструє відкриті етапи(рожевого кольору), ще не відкриті етапи(сірого кольору) та поточний крок(червоний колір).

Рисунок 26 – Trip сторінка

На кожному кроці (див. рис.27) ми маємо 2 кнопки завдяки яких ми можемо перейти до наступного або попереднього кроку. Також ми можемо переходити між етапами завдяки progress-bar. На останньому кроці (див. рис.28) ми маємо вітання від анімованого котика-програміста про закінчення курсу та кнопку, яка переадресовує нас до розділу “Tasks”.

Рисунок 27 – Trip сторінка кнопки

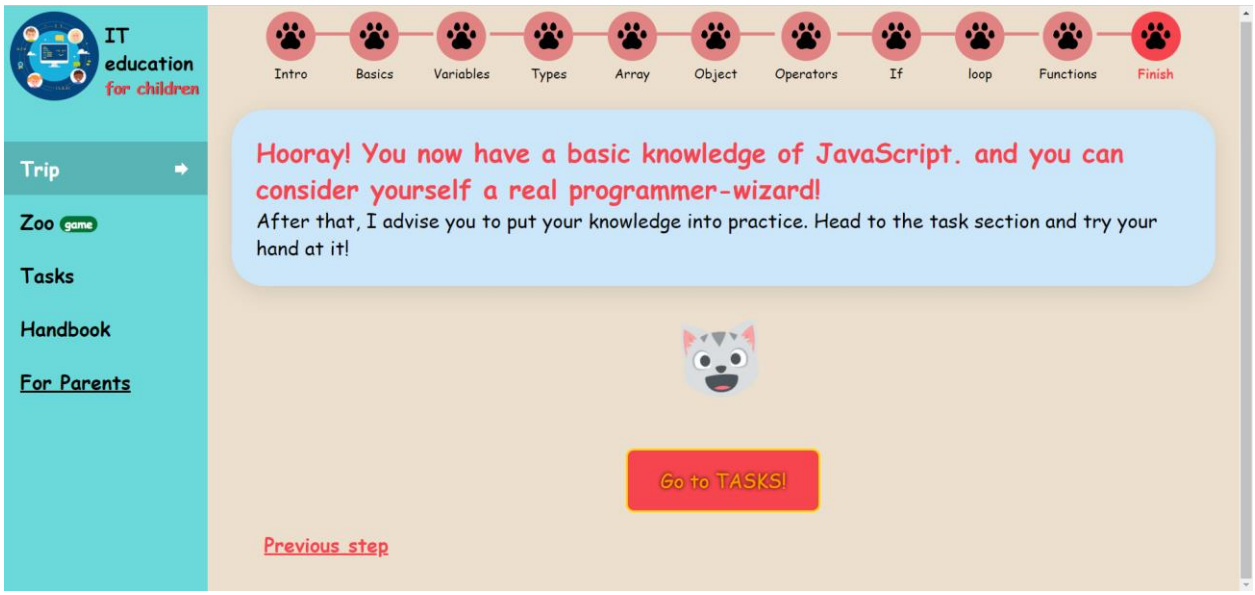

Рисунок 28 – Trip остання сторінка

Переходимо на розділ “Tasks” (див. рис.29). Цей розділ – це різні завдання, які показуються нам по одному. Завдання є як і з можливостю вибору відповіді як і розгорнутою відповіддю. На сторінці завдання ми маємо три кнопки.

Рисунок 29 – Tasks сторінка

Анімована кнопка з котиком “Help” – показує підказку із правильною відповіддю на завдання (див. рис.30).

Рисунок 30 – Tasks сторінка

“Check answer” - перевіряє правильність обраної відповіді. Якщо відповідь не вірна - отримуємо червоне сповіщення про це (див. рис.31).

Та якщо відповідь правильна – отримуємо зелене сповіщення про це, і кнопка “Next task” стає красного кольору і ми можемо перейти до іншого завдання(див. рис.32).

Рисунок 31 – Wrong answer

Рисунок 32 – Right answer

Переходимо на розділ “Zoo” (див. рис.33-34). Це розділ-гра, при відкриванні цієї сторінки ми спочатку отримуємо сповіщення з інформацією, що це за гра, яка її мета та як потрібно грати.

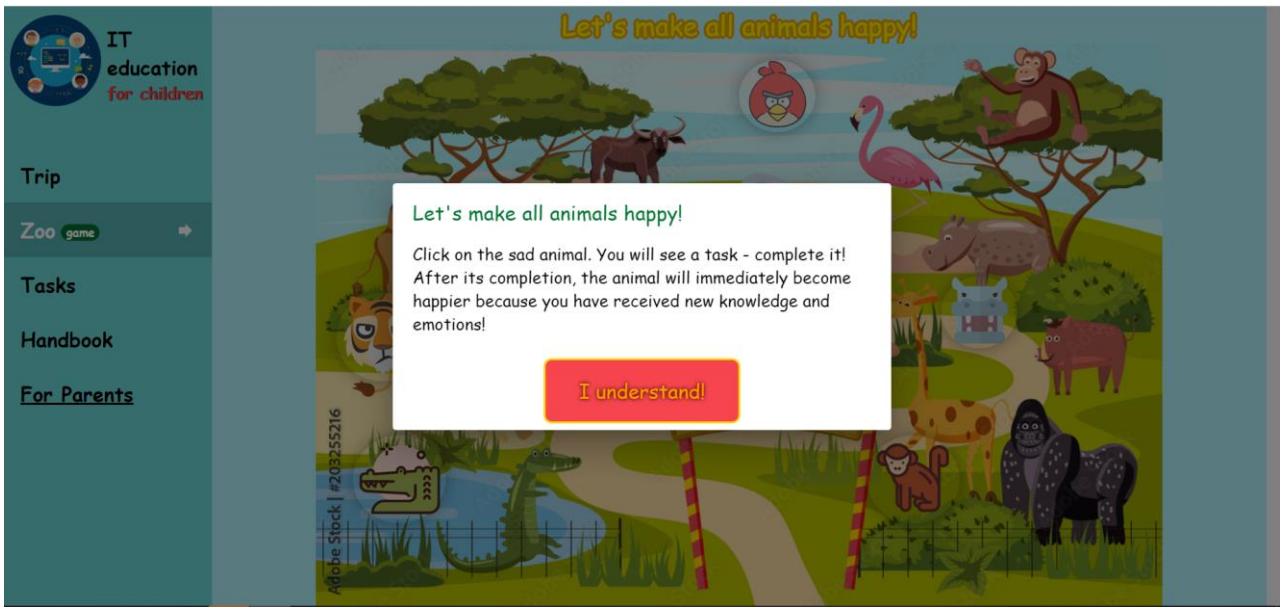

Рисунок 33 – Zoo modal

Рисунок 34 – Zoo

Мета цієї гри - проходячи різні завдання, зробити всіх тваринок щасливими. Сумні тваринки-анімовані і при кліку на неї з'являється вікно із завданням тваринки (див. рис.35). А після виконання завдання тваринка стає щасливою (див. рис.36).

Рисунок 35 – Animal task

Рисунок 36 – Happy animal

Деякі тваринки мають завдання - тести на перевірку знань з JS, а деякі мають грайливі завдання (див. рис.37).

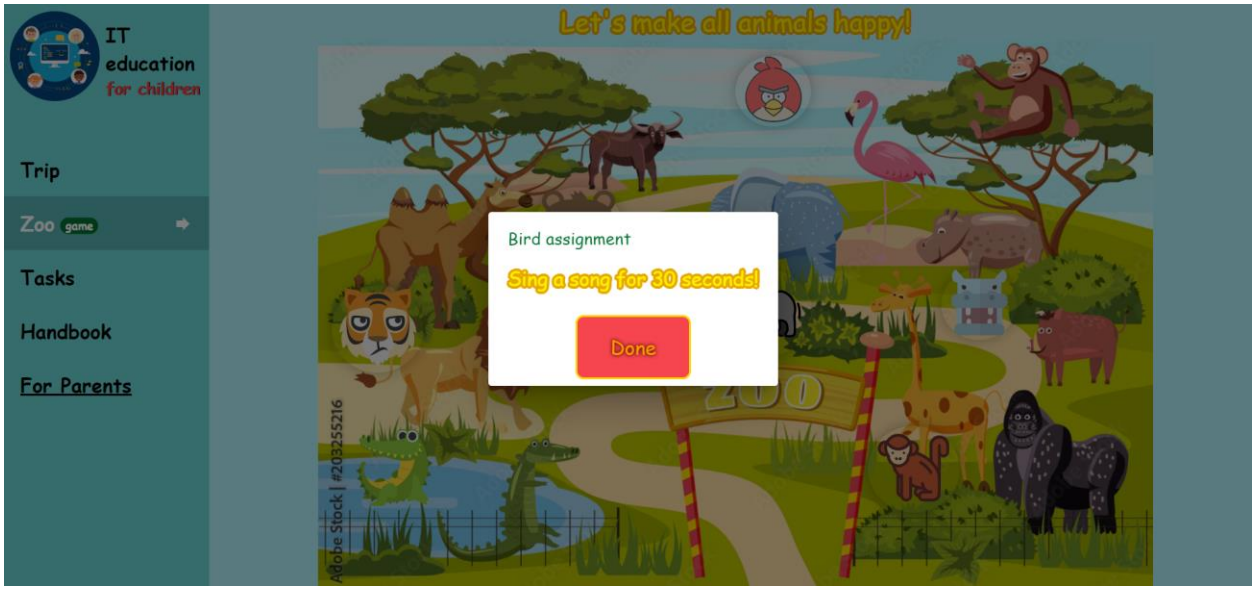

Рисунок 37 – Animal funny task

А коли всі тваринки стають щасливими – гра закінчена, і ми отримуємо відповідний надпис (див. рис.38).

Рисунок 38 – All happy animals

Переходимо на розділ “Handbook”(див. рис.39-40). В цьому розділі ми маємо зручно викладений базовий синтаксис розділений по темам.

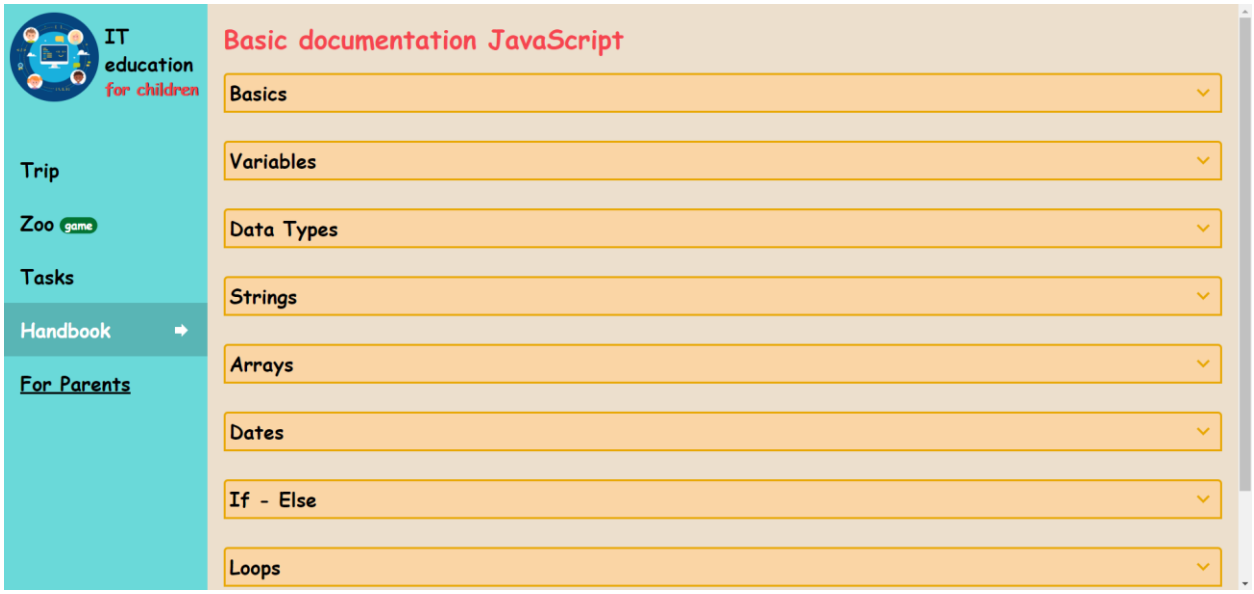

Рисунок 39 – Handbook

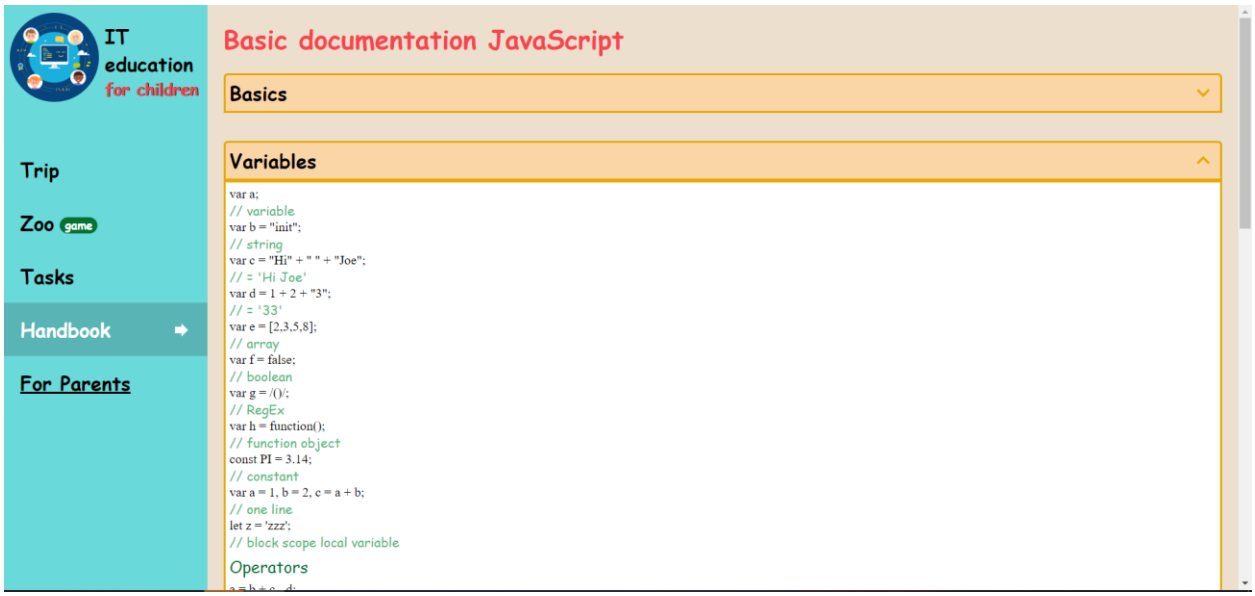

Рисунок 40 – Handbook відкритий розділ

Переходимо на розділ “For parents” (див. рис.41). На цій сторінці ми бачимо відображену інформацію для батьків, де зазначено що це за веб додаток, та мета його створення .

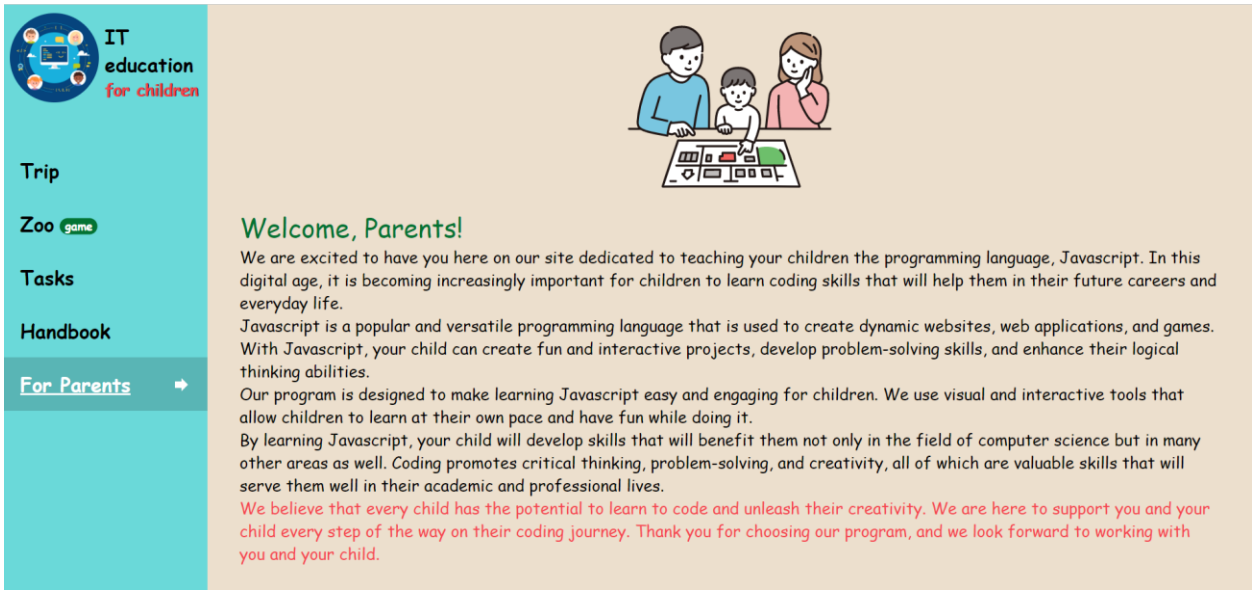

Рисунок 41 – For parents

ВИСНОВКИ

Під час розробки веб-додатку досліджено процес навчання дітей основам мови програмування JavaScript. Проаналізовано існуючі додатки навчання різним мовам програмування, визначено переваги та недоліки кожного з них, визначено аспекти для покращення та поставлено відповідні задачі.

Спираючись на поставлені задачі, було сформовано вимоги до дизайну та функціоналу застосунку.

Створено веб-додаток навчання JavaScript для дітей, який включає в собі декілька спеціалізованих розділів. Веб-додаток має сторінки, на яких є теоретична інформація, тестові завдання для перевірки навичок у програмуванні та гра, граючи в яку можна закріплювати отриманні знання.

За підсумками розробки проекту, результатом роботи є повністю працездатний веб-застосунок, який допоможе дітям отримати базові знання та навички з JavaScript.

Цей проект є актуальним і перспективним, оскільки він відповідає сучасним тенденціям та змінам в світі. І саме тому побудовано чіткі плани на майбутнє, спрямовані на подальший розвиток та вдосконалення веб-додатку. Планується створення можливості для дітей обговорювати свої успіхи та ділитися своїми проектами. Також планується надання можливості батькам та вчителям стежити за прогресом дітей і давати зворотний зв'язок. Забезпечення доступ до ресурсів та порад для батьків та вчителів, щоб вони могли підтримувати навчання своїх дітей.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. JavaRush [Електронний ресурс] – Режим доступу до ресурсу: https://javarush.com/.
2. CodeCademy [Електронний ресурс] – Режим доступу до ресурсу: https://www.codecademy.com/.
3. React [Електронний ресурс] – Режим доступу до ресурсу: https://react.dev.
4. Material UI [Електронний ресурс]. – Режим доступу до ресурсу https://mui.com/.
5. Yarn [Електронний ресурс]. – https://yarnpkg.com/
6. Designing websites for kids [Електронний ресурс]. – Режим доступу до ресурсу https://www.canva.com/learn/kids-websites/
7. Web Design for Kids [Електронний ресурс]. – Режим доступу до ресурсу https://www.rasmussen.edu/degrees/design/blog/web-design-for-kids/
8. How to Teach Coding to Kids [Електронний ресурс]. – https://codevidhya.com/teach-coding-kids
9. W3C [Електронний ресурс]. – https://www.w3.org/
10. Morgan N. JavaScript for kids / N. Morgan., 2015 – 337 с.
11. Myers M. A Smarter Way to Learn JavaScript / M. Myers., 2014 – 288 с.
12. Demirov I. Learn JavaScript VISUALLY / I. Demirov., 2014 – 82 с.
13. CodeMonkey [Електронний ресурс]. – https://www.codemonkey.com/

ДОДАТКИ

Додаток А. Kids Website Color Schem