6COSC023W

Computer Science Final Project

Final Year Project (FYP) - Report

Brainwave

Student: Pakapon Pakdesupaphol (w1909755)

Supervisor: Nagandiram Sadeseevam

Degree: BSc Software Engineering degree

This report is submitted in partial fulfillment of the requirements for the

BSc (Hons) Computer Science degree

BEng Software Engineering degree

School of Computer Science & Engineering

University of Westminster

Date: 25/3/2025

Document Scope

The purpose of this document is to describe and reflect on the processes that took place in developing the Final Project. Discuss any ethical issues associated with your project and describe the methodology that was adopted to develop its design, implementation and testing.

All chapter word counts in this document are approximate and are not intended to be prescriptive.

Declaration

This report has been prepared based on my own work. Where other published and unpublished source materials have been used, these have been acknowledged in references.

Word Count: 12,542
Student Name: Pakapon Pakdesupaphol
Date of Submission: 05/05/2025

Abstract

This project addresses a critical issue in digital education: the limited engagement and interactivity in e-learning platforms for primary-level students. Research shows that younger learners require stimulating environments to stay engaged, and traditional e-learning often relies on static, text-heavy formats that can hinder comprehension and motivation.

The project's primary aim was to design an interactive e-learning platform tailored to the needs of young learners. The platform was developed to deliver educational content while engaging students through intuitive design and interactive features. An essential addition was a tutor-finding tool to facilitate access to support when needed.

To achieve these goals, the project employed front-end and back-end web development techniques, integrating features like quizzes, games, and visual cues to promote active learning. HTML, CSS, JavaScript, and PHP were utilised to build a responsive and user-friendly platform that is accessible across devices.

The final outcome is a fully functional e-learning platform that combines beginner-friendly educational content with engaging features. Results indicate that interactive components significantly enhanced user engagement and enjoyment in learning. A key feature allows users to track their progress and revisit lessons, ensuring a comprehensive learning experience. In conclusion, the project successfully delivered an effective and engaging digital learning solution for primary-level students.

Acknowledgements

I would like to express my sincere gratitude to my supervisor, Nagandiram Sadeseevam, for their invaluable guidance, continuous support, and constructive feedback throughout this project. Their expertise and encouragement greatly contributed to the success of this work.

I also extend my thanks to the lecturers and teaching staff in the Sharan Bhambra and Simon Courtenage programme for providing the foundational knowledge and inspiration that shaped the development of this application.

I am grateful to my classmates and peers for their helpful discussions, suggestions, and feedback during the development and testing stages.

Special thanks to the individuals who participated in testing the platform, both expert and non-expert users, whose input was essential in refining the user experience.

Finally, I would like to thank my family and friends for their patience, motivation, and unwavering support throughout the duration of this project.

Table of contents

List of Figures

List of Tables

1. Introduction

This project aims to enhance e-learning platforms to be more user-friendly and engaging for younger users. The goal is to create an environment where children can easily access content, engage in interactions, and enjoy the learning process. Existing e-learning platforms often focus on adult learners, presenting dense, text-heavy materials that can overwhelm younger users.

Younger learners frequently struggle to maintain their engagement with these platforms due to a lack of visually stimulating content, interactive features, and relatable narratives. Thus, our vision is to develop an e-learning platform emphasising age-appropriate content and interactive learning experiences through games, quizzes, and hands-on activities. This will transform learning into an enjoyable adventure rather than a mundane task, with features like progress trackers and rewards to motivate students.

Content will be crafted using simple language and clear explanations, breaking complex topics into manageable segments. By integrating storytelling and relatable characters, we can help children connect with the material meaningfully. Our platform will feature vibrant visuals and a user-friendly interface to reduce frustration.

Gamified elements will enhance engagement, allowing learners to earn rewards, advance levels, and participate in challenges. Feedback mechanisms and personalised learning paths will ensure that every child feels supported. Furthermore, peer collaboration through group challenges and forums will encourage a sense of community.

In summary, this project aims to fill the gap in e-learning for younger learners by focusing on engagement, interaction, and enjoyment. By prioritising the learner's experience, this platform will present age-appropriate content and gamified elements, allowing children to enjoy their educational journey while achieving their goals.

1.1 Problem statement

E-learning platforms have transformed education by offering convenience and accessibility; however, they often fall short in supporting younger learners. A key issue is that content is frequently too advanced, assuming prior knowledge that young students do not possess, leading to disengagement and comprehension difficulties. These platforms are primarily designed for adult learners, with linear, text-heavy formats that do not align with children's dynamic learning styles.

Educational theories, such as Vygotsky's Zone of Proximal Development (ZPD), emphasise the need for scaffolding to help learners bridge gaps in their understanding. Unfortunately, many e-learning platforms overlook these concepts, presenting content unsuitable for younger users, who benefit more from hands-on and interactive experiences. Mayer's Cognitive Theory of Multimedia Learning suggests that students retain information better when it includes visuals and interactivity, yet many platforms still focus on dense content, resulting in cognitive overload.

E-learning platforms must adopt age-appropriate language, vibrant visuals, and interactive components to improve engagement. Research shows that gamification and immediate feedback significantly enhance motivation among children. However, many existing platforms prioritise content volume over quality, overwhelming students and diminishing their interest.

Additionally, the visual design needs to cater to younger learners. While minimalist interfaces work for adults, children require colourful and engaging elements. Platforms that integrate dynamic designs, compelling narratives, and interactive modules can create a more effective learning experience. By focusing on the needs of younger learners, e-learning platforms can provide enjoyable and practical educational experiences that enhance understanding and retention.

1.2 Aims and Objectives

The goal of this project is to enhance e-learning platforms to better support younger users by creating a learning experience that is more engaging, interactive, and enjoyable. Many current e-learning platforms are designed primarily for advanced learners and adults, often presenting complex content that younger users find difficult to comprehend. Additionally, these platforms usually lack interactive features, which can make learning less engaging and may discourage younger students. This project aims to address this issue by developing an e-learning platform that integrates gamified learning elements, interactive content, and visually appealing illustrations. The objective is

establishing an environment that simplifies educational content while promoting active participation, ultimately making learning effective and enjoyable. By enhancing user engagement and comprehension, this platform will provide younger learners with an educational experience tailored to their specific needs and learning styles.

To achieve this vision, several key objectives will be pursued. First, the platform will offer age-appropriate content that enables younger users to easily grasp and retain information without feeling overwhelmed. Second, gamified elements, such as interactive quizzes, rewards, and challenges, will be incorporated to motivate students and maintain their interest in learning. Furthermore, visual enhancements, including illustrations and animations, will be added to support comprehension and make the learning process more engaging. The platform will also feature a user-friendly interface designed specifically for younger learners, ensuring that navigation is intuitive and straightforward. In addition, adaptive learning techniques will be employed to customise the learning experience according to each student's progress and needs. This platform will incorporate features that facilitate customisation for both administrators and users. Additionally, it will enable administrators to manage user accounts, thereby enhancing organisational oversight effectively. Users will be provided with interfaces that facilitate the editing and customisation of their account details.

2. Background

E-learning is now a key aspect of modern education, especially for younger primary school students. Popular platforms like Khan Academy Kids and ABCmouse use videos, animations, and gamification to engage learners. However, research shows limited interactivity and personalised support can hinder long-term engagement and comprehension (Smith & Lee, 2020).

Developers employ technologies such as HTML, CSS, and JavaScript to improve these platforms for interactive user interfaces, while PHP supports back-end features like user authentication and progress tracking (Rahman & Gupta, 2020). Tools like Visual Studio Code are favoured for their versatility and debugging capabilities (Kumar, 2021).

Gamification elements—like quizzes and badges—enhance motivation and retention. Research by Jones and Brown (2019) demonstrates that gamified e-learning environments boost cognitive engagement and improve learning outcomes.

Despite these advances, platforms are still needed to provide beginner-friendly content and highly interactive experiences for young learners. This project aims to bridge that gap by creating an engaging e-learning website tailored to primary-level students.

2.1 Literature survey

The initial literature survey for this project focused on interactive e-learning platforms targeting young learners, particularly in primary schools. It examined existing platforms, relevant research, and web development tools that foster engaging, learner-centred digital experiences. The aim was to understand how interactive features and content presentation influence learner engagement and educational outcomes.

E-learning platforms like Khan Academy Kids, ABCmouse, and Duolingo for Kids have gained popularity, especially during the COVID-19 pandemic (World Bank, 2020). These platforms use multimodal learning, combining text, images, and gamification to support literacy and general knowledge (Mayer, 2014). However, a common challenge is the lack of personalisation for varying abilities, leading to disengagement when content becomes repetitive or too complex (Smith and Lee, 2020; Jones and Brown, 2019). Researchers advocate for more adaptive environments that provide real-time feedback.

From a technical perspective, essential tools include HTML, CSS, and JavaScript for front-end development, creating responsive and interactive user interfaces (Anderson, 2021). Meanwhile, PHP is commonly used in the back end for managing databases and user tracking (Rahman and Gupta, 2020). These technologies allow for personalised learning experiences and progress monitoring for parents and educators.

Practical development tools also play a key role; Visual Studio Code is often preferred for its versatility and extensive support for languages and frameworks (Kumar, 2021). Gamification is another crucial theme, with studies showing that game elements significantly enhance engagement and motivation (Deterding et al., 2011; Jones and Brown, 2019). Platforms like Duolingo and ABCmouse use rewards to encourage consistent learning.

Overall, the review indicates many tools and methods exist for developing interactive e-learning platforms, although there is still a need for better personalisation and adaptation to individual learner needs.

2.2 Review of projects/applications

Creating an interactive e-learning platform for primary-level students necessitates a comprehensive understanding of pedagogical principles and technical frameworks. This background research delves into current applications and technologies pertinent to the project, evaluating their features, methodologies, and overall impact on learning outcomes.

Additionally, it contrasts the tools and techniques employed in existing systems with those chosen for this project, specifically HTML, CSS, JavaScript, PHP, and the Visual Studio Code (VS Code) development environment.

Existing Projects and Applications

Numerous notable e-learning platforms have been specifically developed for children aged 5 to 11. Khan Academy Kids, ABCmouse, and Duolingo for Kids stand out among these. Each platform presents unique features for early learning, ranging from interactive exercises to adaptive feedback systems. The table below compares the primary features of these platforms with those of the proposed project.

Feature	Khan Academy	ABCmouse	Duolingo	Proposed Project
Age Group Targeted	2–8 years	2–8 years	4–10 years	6–12 years
Gamification Elements	Yes	Yes	Yes	Yes
Visual Interactivity	High	High	Medium	High
Content Adaptation	Limited	Moderate	Moderate	Dynamic (future)
Tutor/Assistance Integration	No	Yes (limited)	No	Yes (custom tutor finder)
Account/Profile System	Yes	Yes	Yes	Yes
Backend Interaction Tracking	Yes	Yes	No	Yes (PHP/MySQL)

Table 1.Comparison between the background research

Analysis of Existing Applications

Khan Academy Kids is designed around vibrant characters and engaging voiceovers that guide children through various activities. It incorporates basic interactivity, touchable images and audio prompts, but lacks personalisation and tutor-based support (Khan Academy, 2023).

ABCmouse offers a structured curriculum in reading, math, and science. It features user profiles and progress tracking; however, it is subscription-based and therefore less accessible to lower-income households (ABCmouse, 2023). Additionally, its content is static and does not adapt in real time.

Duolingo for Kids leverages repetition, translation, and voice recognition. While it effectively employs gamification, its limited subject range renders it unsuitable as a comprehensive learning tool for primary education (Duolingo, 2022).

In contrast, the platform developed for this project seeks to integrate the strengths of these systems while addressing their limitations. Key innovations include a streamlined content structure, interactive quizzes and mini games, along with a feature that matches students with available tutors for more guided support.

Tools, Frameworks, and Methods Utilised

This project employs a standard web development stack comprising HTML, CSS, JavaScript, and PHP.

- HTML (Hypertext Markup Language) is the foundation for structuring web content. It allows for creating learning modules, buttons, input fields, and containers for text and images.
- CSS (Cascading Style Sheets) adds design styling, encompassing layouts, colours, and hover effects, enhancing the visual appeal of the platform for children.
- JavaScript facilitates dynamic behaviour, enabling features such as quiz answer validation, content visibility toggling, and responsive user interactions like clicking or dragging objects. These interactive elements are essential for maintaining engagement among young learners.
- PHP drives the server-side logic, handling user authentication, tracking quiz results, and providing connectivity to the Mysql database, which stores user progress and preferences.

```php
<?php
session_start();

    include('function.php');
    include('connect.php');

    $error_message = "";
    $show_error = false;

    if($_SERVER['REQUEST_METHOD'] == "POST")
    {
        $username = isset($_POST['username']) ? $_POST['username'] : '';
        $password = isset($_POST['password']) ? $_POST['password'] : '';

        if(!empty($username) && !empty($password))
        {
            if (!$con) {
                die("Connection failed: " . mysqli_connect_error());
            }

            $query = "SELECT * FROM registration WHERE username = ?";
            $stmt = mysqli_prepare($con, $query);
            mysqli_stmt_bind_param($stmt, "s", $username);
            mysqli_stmt_execute($stmt);
            $result = mysqli_stmt_get_result($stmt);

            if($result && mysqli_num_rows($result) > 0)
            {
                $user_data = mysqli_fetch_assoc($result);
                if($user_data['password'] === $password)
                {
                    $_SESSION['user_id'] = $user_data['user_id'];
                    header("Location: index.php");
                    exit();
                }
                else
                {
                    $error_message = "Incorrect username or password";
                    $show_error = true;
                }
            }
            else
            {
                $error_message = "Incorrect username or password";
                $show_error = true;
            }
        }
    }

?>
```

Figure 1. Part of the login PHP

```html
<!doctype html>
<html>
<head>
<meta charset="UTF-8">
<title>Login</title>
<link rel="stylesheet" href="styles/login.css">
<link rel="stylesheet" href="styles/navbar.css">
</head>

<body>
    <nav class="navbar">
        <div class="nav-brand">Brainwave</div>
        <ul class="nav-links">
            <li><a href="index.php">Home</a></li>
            <li><a href="english.html">English</a></li>
            <li><a href="mathciamtic.html">Mathciamtic</a></li>
            <li><a href="user.php">Account</a></li>
            <li><a href="support.html">Support</a></li>
        </ul>
    </nav>

    <div class="wrapper">
        <form method="post" id="loginForm">
            <h1>Login</h1>

            <div class="error-message" id="errorMessage">
                <?php echo $error_message; ?>
            </div>

            <div class="input-box">
                <input type="text" name="username" placeholder="Username" required>
            </div>

            <div class="input-box">
                <input type="password" name="password" placeholder="Password" required>
            </div>

            <div class="remember-forgot">
                <a href="forgetpassword.html">Forgot password</a>
            </div>

            <input type="submit" class="login-btn" value="Login">

            <div class="register-link">
                <p>Don't have an account? <a href="register.php">Register</a></p>

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

Figure 2. Part of the login HTML

Figure 3. Part of the login JavaScript

Visual Studio Code (VS Code) has been chosen as the primary development environment due to its lightweight design, built-in terminal, Git integration, and compatibility with HTML, CSS, JavaScript, and PHP—all within a single platform (Kumar, 2021). Extensions like “Live Server” and “PHP Intelephense” further enhance coding and testing efficiency.

Advantages and Disadvantages of Selected Tools

Tool/Language	Advantages	Disadvantages
HTML/CSS	Easy to learn, essential for structure/design	Limited interactivity alone
JavaScript	Enables interactive experiences	Can be complex for real-time updates
PHP	Widely supported, good for dynamic content	Less modern than Node.js alternatives
VS Code	User-friendly, extensive extension support	Requires setup of local server for PHP testing

Table 2. Advantages and Disadvantages of the tools

These tools were selected for their simplicity, accessibility, and compatibility with educational web platforms. In contrast to frameworks like React or Angular, which offer greater power but come with increased complexity, this simpler stack ensures that the system remains lightweight and maintainable, which are crucial considerations when working with younger learners and limited hardware resources (Anderson, 2021).

Methods and Educational Techniques

This project employs gamification, scaffolding, and interactive learning design as foundational educational strategies. As Deterding et al. (2011) outlined, gamification fosters engagement through elements such as badges, points, and progress levels. Scaffolding, rooted in Vygotsky's theory (1978), enables learners to incrementally build their knowledge with the support of feedback and hints.

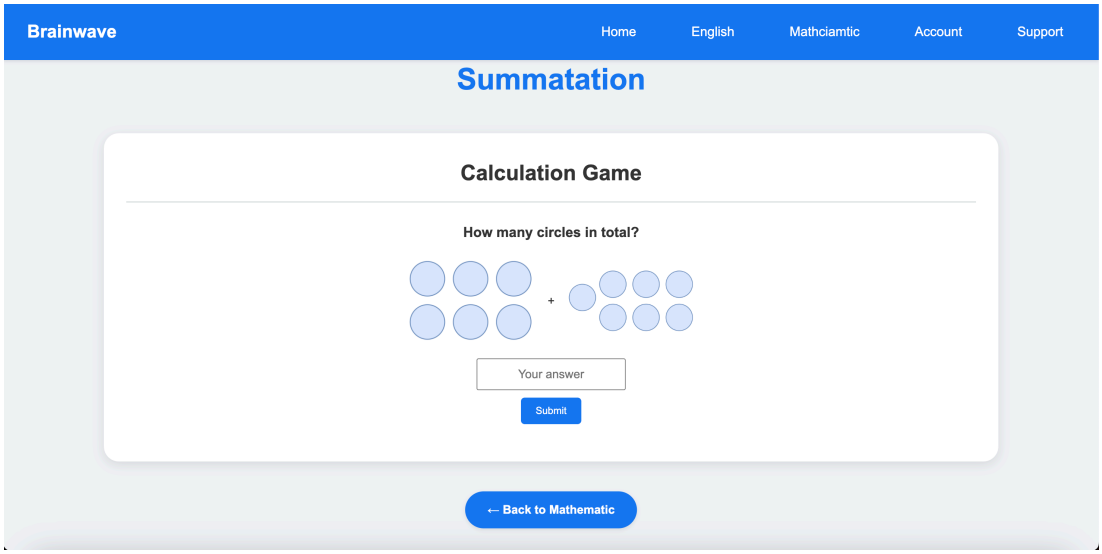

Figure 4. The illustration for the gamification interactions

To accommodate diverse learning styles, interactive quizzes, dynamic hints (such as hover popups), and visual/audio feedback have been integrated into the system. These features ensure that the platform caters to the cognitive needs of young users without overwhelming them.

Conclusion of Comparative Analysis

This research underscored both the strengths and weaknesses of current e-learning platforms, guiding the design decisions for this project. While established systems such as Khan Academy Kids and Duolingo exhibit a high level of polish, they frequently fall short in terms of adaptability and personalised learning support. The proposed platform seeks to bridge this gap by incorporating tutor connection features, enhancing user interactivity, and adopting a more modular design. Backed by proven technologies and effective pedagogical methods, this project presents a scalable and impactful solution for advancing digital learning in early education.

2.3 Review of tools, frameworks and techniques

In developing an interactive e-learning platform for primary-level learners, we examined various tools, programming languages, and libraries to create an accessible and interactive system. This evaluation highlighted the strengths and limitations of commonly used development tools.

HTML was used to structure the website content, providing a straightforward layout for learning modules and interactive components. Although widely supported, HTML lacks dynamic interaction capabilities, necessitating scripting languages.

CSS enhanced the visual design and user interface, engaging young learners through features like hover animations and responsive layouts. Its modular structure improves code maintainability, though implementing complex layouts can pose challenges.

JavaScript added interactivity and real-time updates, which are vital for features like quiz validation and drag-and-drop elements. Its vast library ecosystem, including jQuery, accelerates development but can complicate debugging, especially when using asynchronous functions.

For server-side development, PHP was chosen for its ease of integration with HTML and databases, facilitating user authentication and data management with a Mysql database. Despite its strengths, PHP is sometimes criticised for being less modern compared to technologies like Node.js.

Due to its security and performance, Mysql was used to store user data, though larger platforms might prefer Nosql databases like Firebase for better scalability.

Visual Studio Code (VS Code) was selected as the development environment for its support for various languages and helpful extensions. While it boosts productivity with features like Live Server and PHP IntelliSense, initial setup for specific extensions can be cumbersome.

Comparative Summary of Development Tools

Tool/Framework	Purpose	Advantages	Disadvantages
HTML/CSS	Web structure and styling	Easy, essential responsive and commonly used	Limited interactivity
JavaScript	Client-side interaction	Real-time features interactive UI	Complex for beginners
VS Code	IDE/Editor	Fast for small apps, good integration	Outdated in modern lard-scale use
Bootstrap	UI framework	Quick design, responsive elements	Require setup for the backend testing
jQuery	JaScScipt utility	Easy DOM anent handling	Reduces uniqueness
MySQL	Database management	Reliable, secure, good with PHP support	Limited scalability

Table 3. Comparative Summary of Development Tools

In addition to core programming languages and environments, the project evaluated the utility of various frameworks and libraries capable of accelerating development. Bootstrap, a front-end CSS framework, was assessed for its capacity to produce responsive, mobile-first designs. Its grid system and pre-built UI components can save significant time during interface development. However, one limitation of Bootstrap is the potential for web pages to appear uniform or generic unless accompanied by additional customisations. Additionally, jQuery, a succinct and efficient JavaScript library, was considered for its ability to simplify HTML document traversal, event handling, and AJAX requests. While jQuery excels in rendering JavaScript code more accessible and readable, it is worth noting that the evolution of native JavaScript has diminished the necessity for jQuery in modern web development.

The overall toolset employed, which encompasses HTML, CSS, JavaScript, PHP, Mysql, and Visual Studio Code, was selected based on criteria of accessibility, ease of learning, and demonstrated effectiveness for constructing interactive educational systems. These technologies exhibit high compatibility and support across various devices and browsers, enabling an inclusive and engaging learning environment.

3. Legal, social, sustainability and ethical issues

When developing an e-learning platform tailored for primary-level students, it is imperative to address a diverse array of legal, ethical, social, professional, sustainability, and security considerations to ensure the platform operates within a framework of responsibility, inclusivity, and adherence to best practices. Given the inherently sensitive nature of working with children, strict compliance with legal statutes such as the General Data Protection Regulation (GDPR) within the European context and the Children's Online Privacy Protection Act (COPPA) in the United States becomes vital. These legislative frameworks delineate the parameters governing the collection, storage, and processing of minors' personal information, necessitating explicit parental consent and the implementation of secure data management protocols (Voigt and Von dem Bussche, 2017). Non-compliance with these regulations may incur legal penalties and severely undermine the platform's credibility.

Ethical considerations assume a critical role in creating digital learning environments for children. It is essential to maintain transparency in data management, eschew manipulative design practices, and offer learning materials in an unbiased and developmentally appropriate manner. The ethical principle of fairness stipulates that all learners, irrespective of socio-economic status or physical capabilities, should have equitable access to educational resources (Beauchamp and Childress, 2019). Accordingly, the platform must incorporate accessibility features and culturally inclusive content, consciously avoiding language or imagery that could marginalise any demographic group.

Socially, the platform possesses the potential to profoundly influence learners' perceptions of education and their engagement with digital interactions. As posited by Selwyn (2016), digital technologies in education should serve to foster social inclusion and equip learners with the competencies required to thrive within contemporary society. This undertaking demands careful consideration of content sensitivity, representation of diversity, and the design of user interactions. Additionally, ensuring compatibility with low-cost devices and mobile networks addresses the pervasive digital divide, particularly within under-resourced communities (UNESCO, 2021).

Professional responsibilities inherent to software development must be upheld with diligence. In accordance with the British Computer Society Code of

Conduct (BCS, 2023), professionals are expected to operate with integrity, maintain competency, and prioritise the public interest. This necessitates adherence to established software development standards, meticulous documentation, rigorous usability testing, and transparency regarding the platform's functionalities and limitations. Engaging in a feedback loop with educators and learners throughout the development process is essential to ensure that the resultant product is practical and tailored to its intended users.

From a sustainability perspective, digital platforms can engender positive outcomes by reducing reliance on printed materials and minimising the carbon emissions associated with traditional educational methodologies. Nonetheless, developers must also account for the environmental implications of hosting services and data transmission. Adoption of sustainable software engineering practices, such as efficient coding, low-bandwidth design, and utilisation of eco-friendly hosting solutions, can mitigate the overall environmental impact (Preist, Schien, and Blevis, 2016). These practices align with global sustainability objectives, including those articulated within the United Nations' Sustainable Development Goals (UN SDGs) (United Nations, 2020).

Security emerges as a paramount concern, owing to the youth of the user demographic. Implementing robust encryption protocols, secure authentication mechanisms, and routine vulnerability assessments is essential to safeguard user data. As Anderson (2020) emphasised, an anticipatory approach to cybersecurity is critical, particularly in contexts involving personal and behavioural data. A "privacy-by-design" paradigm should be adopted, limiting data collection to essential information and granting parents comprehensive control over data sharing and account access. Furthermore, the platform should educate users and guardians on safe online practices through instructional materials and alerts.

In conclusion, the development of an e-learning platform for primary learners must transcend mere functional design, rigorously incorporating a thorough examination of legal, ethical, social, professional, sustainability, and security factors. Such an integrative approach ensures compliance with regulatory frameworks and the protection of users and facilitates the establishment of a safe, equitable, and forward-looking digital learning environment.

4. Methodology

Project stage life cycle

This project commenced in mid-October 2025, starting with the Project Proposal and Requirements Specification (PPRS), which includes comprehensive background research conducted at the project's outset. The PPRS outlines the aim, objectives, and scope of the project. As mentioned in this document, the goal is to create an interactive function that engages younger learners, particularly at the primary school level, who often struggle with a lack of motivation and interest in their studies. I aim to provide a platform that makes learning enjoyable and fosters greater engagement among learners, benefiting them in the future.

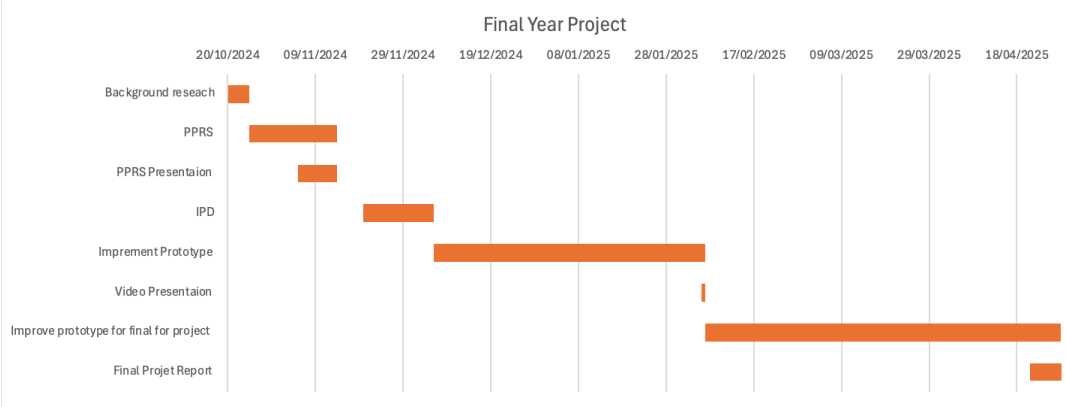

Figure 5. Gantt chart of the schedule of this project at the beginning

After completing my Project Proposal and Requirements Specification (PPRS), I conducted additional background research to delve deeper into each platform. This helped me identify their advantages and disadvantages. I used this information to adapt my project and enhance its content, making it more understandable for younger users while promoting engagement. I discovered that most commonly used platforms focus on self-learning, but my platform will include an option for users to access a tutor when they are struggling.

The main requirement of this project is the interaction with the user and the attraction use with content, with which I have included the JavaScript in this project to make the interaction more interesting to the user, combined with HTML and CSS code, which is the base of the coding the of this website to make the appearance more vibrant like other platform dose, such as Duolingo use colour bright green, ABCmouse use multi colour or khan academy also use green colour. The colour method is significant for memorising the platform.

I have been using VS Code to develop most of the code for my project, as it offers various tools and extensions that help me with coding and hosting the visual website in a browser. This allows me to achieve the precise layout and functionality I want to test. However, VS Code cannot implement a Mysql table independently, so I also use another software called MAMP. MAMP enables me to host the website I'm working on to test functionalities, such as JavaScript, which requires an actual hosting environment to operate. Additionally, I'm working on implementing a Mysql table to support the front-end and back-end functions to manage user details before deploying it on the university server. This process has significantly deepened my understanding of PHP code.

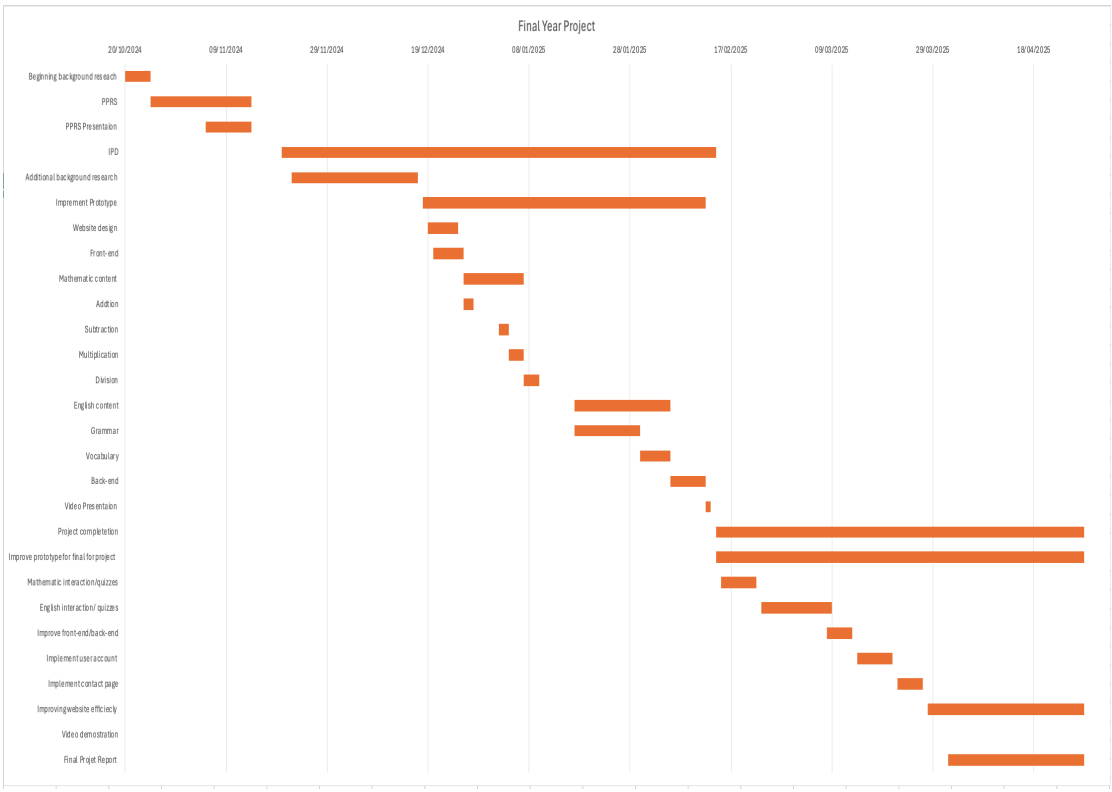

Figure 6. Gantt chart of the schedule of this project, showing when the project is completed

I have designed the interaction and user interface (UI), emphasising simplicity and utilising bright colours to create an engaging experience. This approach aims to ensure that younger users, who are the primary target of this project, can easily recognise and navigate the UI. The interface must be straightforward and uncomplicated, allowing users to understand it effortlessly. Additionally, I have incorporated hover interactions to make the UI more dynamic and engaging. The primary focus is on the images that help the UI look livelier to the user and allow the user to interact.

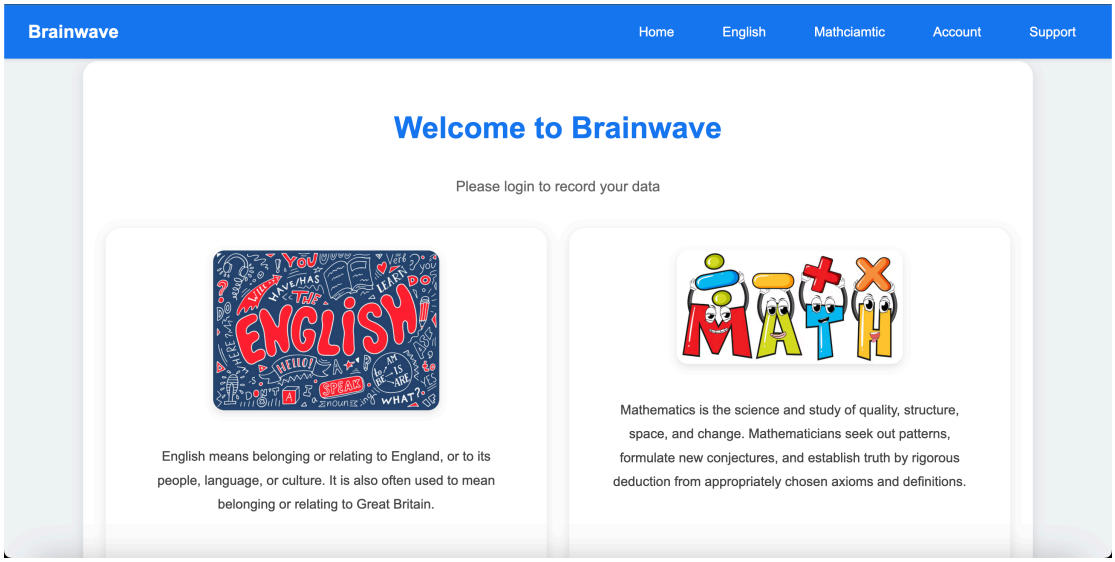

Figure 7. Index page UI

The front-end and back-end components represent the foundational aspects of this project, as they facilitate the website's functionality, user interface, and database operations. For development, I utilised Visual Studio Code as my integrated development environment (IDE), enabling a comprehensive preview of the website before deployment on the server.

The programming languages employed in this project include HTML, CSS, JavaScript, and PHP, each serving distinct purposes. HTML serves as the primary language for structuring content, allowing for the creation of visual elements such as headings, subheadings, and content sections. CSS acts as the stylistic counterpart to HTML, enhancing the aesthetic appeal by incorporating colour, layout design, and interactive features, such as hover effects that bring elements to life and transform square buttons into rounded ones.

JavaScript further enriches the website's interactivity by enabling dynamic functionalities in conjunction with HTML and CSS. I leveraged JavaScript to develop features for the mathematics and English quizzes, which required practical computation and randomisation processes. One aspect of JavaScript that particularly appeals to me is its capability to implement a timer system using straightforward code.

Lastly, PHP introduces a more complex layer of functionality, particularly in its interaction with SQL databases. This language is instrumental in establishing back-end processes, such as user authentication through login pages, as well as managing the display of usernames and other data intricately tied to the database. SQL serves as the primary relational database management system utilised in this project, selected for its familiarity and capacity to support advanced data manipulations and operations.

Website hosting is another crucial aspect that I need to address. I plan to finalise the primary hosting method using the University server at the University of Westminster. Additionally, I will host the website locally on my machine. I

have discovered an application called MAMP, which replicates the server host system and allows me to work with the database system while testing the full functionality of the hosted website. This setup enables me to check the JavaScript functionality and ensure it works well with the HTML and CSS I have previously implemented.

One area I could significantly improve is the PHP functionality within my project, particularly in creating a more robust and scalable connection that would allow for the implementation of additional features. I would be better equipped to develop more complex and reliable systems with more profound knowledge and a stronger understanding of PHP. For instance, I attempted to implement an administrative account feature to facilitate maintenance tasks, such as managing user accounts. However, due to my limited understanding of PHP at the time, the function did not operate as intended, which caused the system to crash upon execution. As a result, I decided to remove the administrative functionality and allow users to delete their accounts if necessary. In the future, enhancing my proficiency in PHP would enable me to successfully implement administrative controls and improve the overall stability and functionality of the platform, leading to a more professional and user-friendly system.

5. Design

Software structure

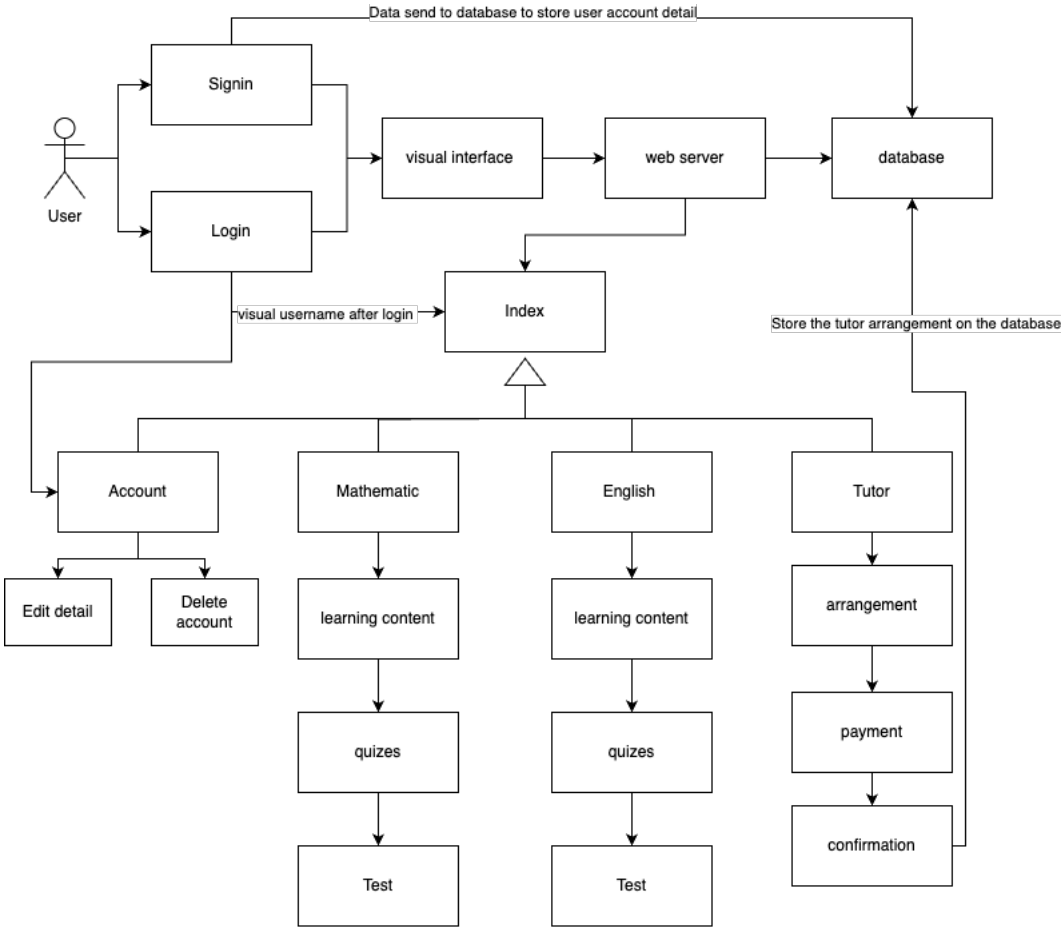

Figure 8. System structure diagram

The software is primarily structured using two combined languages: HTML and CSS. These languages are essential for providing the website's visual interface, allowing users to interact with it. I predominantly use the tag to create sections that enhance the visual layout. Additionally, I utilise tables with <tr> (table row) and <th> (table header) tags to organise necessary content. For lists, I employ <ul> (unordered list) and <li> (list item) tags.

CSS mainly improves the website's appearance, including colour, size, margin, and borders. I often call CSS in the HTML using the class attribute to specify which elements I want to customise. For example, `<div class="content">` indicates I will apply styles from my external CSS file using the ' content` selector. I also frequently use the hover effect in CSS to create interactive hover states.

JavaScript is utilised to create interactive features such as quizzes and timers. In this instance, it controls the quizzes and manages the timer. When a user enters the page, the timer begins counting down, and the quizzes are presented randomly, for example;

```
function submitQuiz() {
    let score = 0;
    questions.forEach((q, index) => {
        const userAnswer = document.getElementById(`answer${index}`).value.trim().toLowerCase();
        if (userAnswer === q.answer.toLowerCase()) {
            score++;
        }
    });
    document.getElementById('result').innerText = `You scored ${score} out of ${questions.length}`;
}
```

Figure 9. JavaScript submit function.

This is the submission function. To execute this function, the user needs to answer a question. The answer will be stored in a temporary storage system before proceeding. The function will then check if the answer is correct by comparing the input value to an answer index. It will track whether the response is right or wrong for each question.

PHP is the only language that enables me to connect to a database through the server. This language requires accurate data to establish a direct connection; it will not function correctly if any information is missing. For example, I have configured two connections: one for my local hosting system and another for the university server.

```
<?php

    $dbhost = 'localhost';
    $dbuser = 'add';
    $dbpass = '1234';
    $dbname = 'detail';

    if(!$con = mysqli_connect($dbhost,$dbuser,$dbpass,$dbname))
    {
        die("failed to connect!");
    }
?>
```

Figure 10. PHP local connection

```php
<?php

    $dbhost = 'phpmyadmin.ecs.westminster.ac.uk';
    $dbuser = 'w1909755';
    $dbpass = 'rSNkseWMYi97';
    $dbname = 'w1909755_0';

    if(!$con = mysqli_connect($dbhost,$dbuser,$dbpass,$dbname))
    {
        die("failed to connect!");
    }
?>
```

Figure 11. PHP university connection

As you can see in Figures 10 and 11, the connection details are entirely different, allowing me to connect to a different server, which also allows me to access the back-end implementation, such as storing data collection and management.

User interface (UI)

The main design of this user interface prioritises simplicity, making it suitable for younger users, who are the primary focus of this project. I have enhanced the visibility of the buttons by using a colour that aligns with the website's colour scheme, which is blue. The contrast between the light grey background and the white content div improves visual appeal. Additionally, the use of images adds interest to the UI, allowing users to grasp the content without any confusion quickly. I also included brief explanatory text to give users a hint about their experience on the next page.

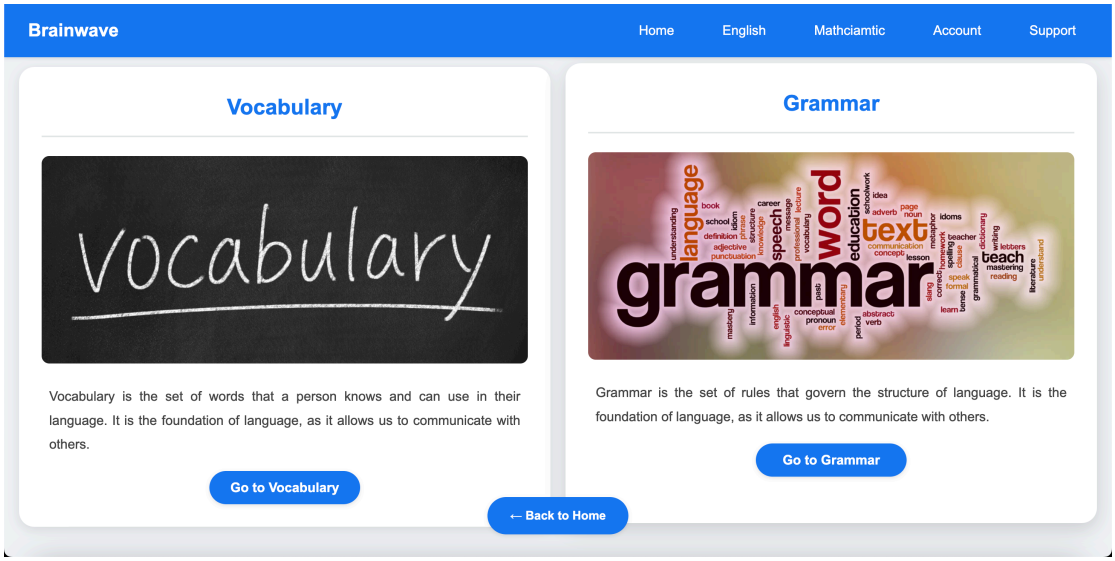

Figure 12. English page UI

The visual elements are created using a combination of HTML and CSS. This allows me to organise the content into different sections with HTML, using CSS

to add colour and other visual features. CSS also helps to smooth the borders and create spacing by using margins.

Infrastructure

The website will be launched on the university server, allowing me to create additional functions and ensure the website's sustainability for an improved user experience. To proceed with this, I need to gain access to the server and create a public folder for my files. Additionally, I will connect the PHP to the server by using phpMyAdmin under my university account to create the necessary Mysql data tables. These tables will store data collected from user inputs, such as user details and tutor arrangement dates and times.

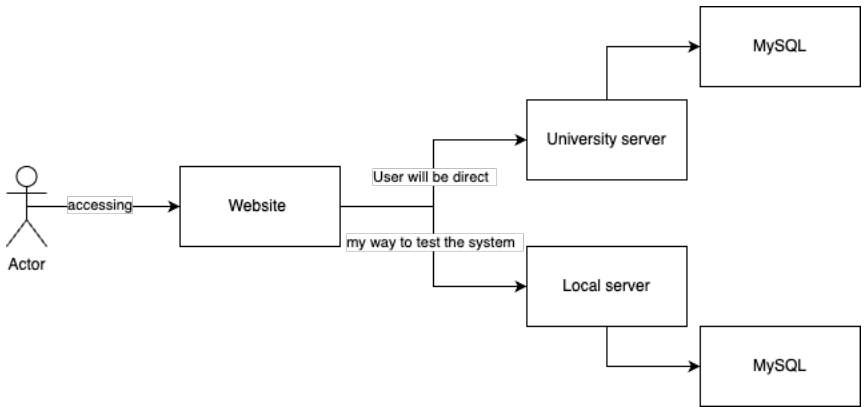

Figure 13. Inference structure diagram

For the local test, I used a software called MAMP, which enables me to set up a local server on my machine. Additionally, MAMP includes its version of phpMyAdmin, allowing me to create and test Mysql databases before deploying them to the university server.

Functionality

The website offers a range of functionalities, including quiz pages for various subjects and additional features such as login and sign-in pages.

The login and sign-in pages are similar in their structure. They each utilise a text box to capture user input, such as usernames and passwords. Upon clicking the submit button, a PHP function processes the input, storing the data in the user database. This enables me to use or edit the information as needed.

The quiz functions are more complex due to the variety of quizzes available on the website. For example, I have implemented a quiz for mathematical subtraction. In this quiz, I use JavaScript to randomly generate two values, ensuring that the first value is always greater than the second. I then subtract the second value from the first and temporarily store the result. After the user provides their answer, the function checks whether it is correct or incorrect,

displaying the correct answer if it is right, or indicating the wrong calculation if the answer is incorrect.

Algorithm development

The algorithm I have is to detect whether the tutor has already been booked at the same time. This part will involve the database and mainly use PHP. PHP allows me to implement a function to check the data in the Mysql table to see if the data already exists. The function will deny processing and redirect the user to the select tutor page to arrange again.

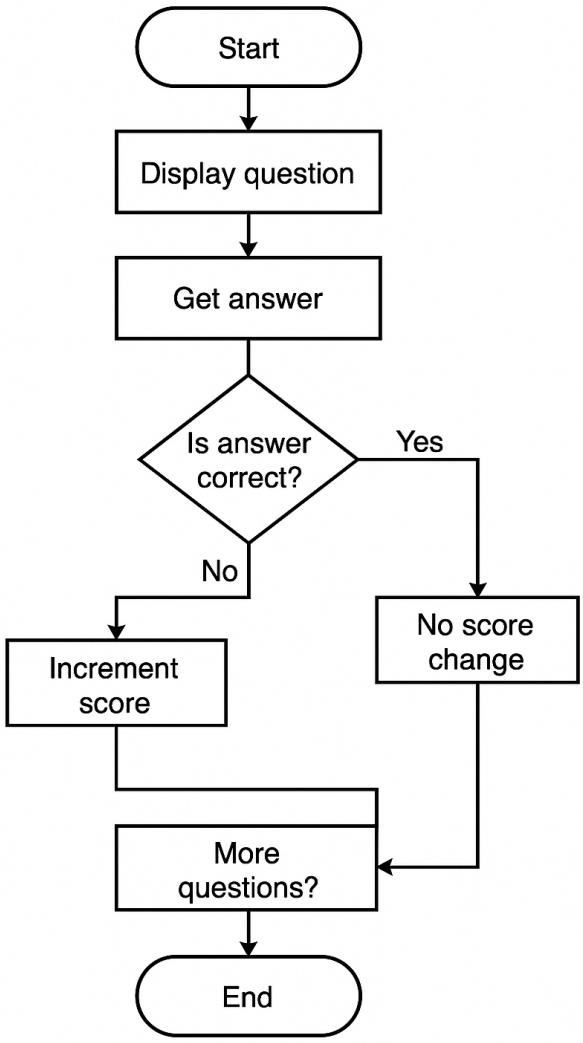

Figure 14: Flowchart of the quiz system.

The quiz will include a scoring feature that operates simultaneously with the main quiz functions. It will first execute the appropriate function based on the type of quiz being taken. While the quiz is in progress, the score will be calculated, allowing for the total number of questions answered and the user's score to be displayed once the quiz is completed.

Content creation

The e-learning platform primarily targets primary school students aged 6 to 12 years. The content is designed to engage this age group and encompasses all topics relevant to their school curriculum. The subjects covered on the website include English and Mathematics, each featuring various subtopics for students to explore or practice during their free time. This approach aims to enhance their skills in these areas, making learning effective and enjoyable. The content I have chosen is primarily designed for primary school students. My focus is on creating a pleasant learning experience that enhances user engagement and facilitates better learning outcomes. To achieve this, I have ensured that the learning website is easy to navigate, featuring simple images that guide users, regardless of age. The content encourages interaction, keeping students engaged with the study materials.

The reliability of the learning content is based on commonly accessed websites, ensuring accuracy. Additionally, the website's performance is compatible with all browsers and platforms, offering users the best experience anytime, anywhere, and on any device. The algorithms used in this project are precise and highly accurate; they generally operate one function at a time, with only a few instances where two functions run simultaneously. This approach enhances overall performance and reduces inaccuracies.

6. Tools and implementation

6.1 Tools

The development of the interactive e-learning platform involved utilising various programming languages, development environments, and web frameworks to create a user-friendly, functional, and engaging website. This section outlines the technologies employed, explains their selection based on the project's requirements, and reflects on the skills applied and newly acquired throughout the development process.

The core structure of the website, including content creation and organisation, primarily utilises HTML and CSS to enhance visual appeal and provide an efficient user experience. JavaScript facilitates user interactions with the website, such as quizzes, tests, and answer validations. Each function serves a specific purpose, primarily aimed at engaging users effectively.

PHP is another essential component of the website’s architecture. It enables it to interact with the database for data validation, such as login functionalities, and facilitates data storage through a Mysql database. The Mysql database plays a crucial role in data management, working with PHP to handle user information and more.

I have also incorporated several frameworks, including Bootstrap, to ensure mobile responsiveness—a critical feature for a modern e-learning platform that

must be accessible on tablets and smartphones. This adaptation guarantees that my platform operates smoothly across various devices. Additionally, I utilised a PHP library called Farmwear, which aids in establishing PHP connections and coding, offering a collection of pre-written programs that streamline the development process for my website. Both frameworks help speed up development, allowing me to use everything while ensuring compatibility with my programs.

Through this project, I have learned PHP, which has deepened my understanding of the backend functions of a website. I now have a clearer insight into the connection between the server and the code I have written. This knowledge enables me to incorporate various creative functions and effectively manage the database, allowing me to store as much data as needed. Additionally, this project has highlighted areas where I needed improvement, motivating me to study and implement the functions I want to include in my platform.

6.2 Implementation

This section provides a comprehensive overview of the technical implementation of the key features of the interactive e-learning platform designed for primary-level students. It includes code explanations presented through essential use cases, each accompanied by pseudocode or annotated code snippets. The discussion will cover the programming languages used in this project, such as HTML, CSS, JavaScript, and PHP, and will briefly describe the front-end and back-end development, as well as the Mysql database. Additionally, the primary functions of the platform, such as registration, login, tutor booking, and payment processing, will be detailed. This will allow me to explain each function in depth.

Registration and login

The purpose of this function is to provide users with secure access to their accounts. It enables users to create a username and password to access their account and track their learning progress. This function encompasses front-end and back-end components and integrates them with a Mysql database.

Here is an example of the pseudocode of the login function.

```
Start session

If form is submitted (POST request):

    Get the username and password from the form
```

If both fields are not empty:

Connect to the database

Query for the user with a matching username

If the user exists:

Check if the password matches

If correct:

Set the session and redirect to the home page

Else:

Show error message

Else:

Show error message

Display login form with:

- Username and password input
- Error message if login fails
- Forgot password and register links

Add JavaScript to:

- Hide error message after a few seconds
- Hide error message when typing starts
- Prevent form resubmission on refresh

```php
<?php
session_start();

include('function.php');
include('connect.php');

$error_message = "";
$show_error = false;

if ($_SERVER['REQUEST_METHOD'] == "POST") {
    $username = isset($_POST['username']) ? $_POST['username'] : '';
    $password = isset($_POST['password']) ? $_POST['password'] : '';

    if (!empty($username) && !empty($password)) {
        if (!$con) {
            die("Connection failed: " . mysqli_connect_error());
        }

        $query = "SELECT * FROM registration WHERE username = ?";
        $stmt = mysqli_prepare($con, $query);
        mysqli_stmt_bind_param($stmt, "s", $username);
        mysqli_stmt_execute($stmt);
        $result = mysqli_stmt_get_result($stmt);

        if ($result && mysqli_num_rows($result) > 0) {
            $user_data = mysqli_fetch_assoc($result);
            if ($user_data['password'] === $password) {
                $_SESSION['user_id'] = $user_data['user_id'];
                header("Location: index.php");
                exit();
            } else {
                $error_message = "Incorrect username or password";
                $show_error = true;
            }
        } else {
            $error_message = "Incorrect username or password";
            $show_error = true;
        }
    }
}
?>
```

Figure 15: Snippet code of the logging function

This section focuses on the crucial data retrieval and validation process, following the simple pseudocode provided earlier. First, it checks the text box to determine whether input has been entered. An error message will appear if the box is empty, prompting the user to enter their registered username and password. Once the inputs are provided, the validation process begins by connecting to the MySQL database. It will verify whether the username and password entered by the user match the stored data. If they are correct, the user will be redirected to the index page. If the password or username is wrong, it will result in an error, and the error message function will be executed to give the user notice and retry again, as shown in Figure 16.

```
<script>
    // Show error message for a few seconds if there is an error
    <?php if ($show_error): ?>
        document.getElementById('errorMessage').style.display = 'block';
        setTimeout(function() {
            document.getElementById('errorMessage').style.display = 'none';
        }, 3000);
    <?php endif; ?>

    // Hide error message when user starts typing
    document.querySelectorAll('input').forEach(input => {
        input.addEventListener('input', function() {
            document.getElementById('errorMessage').style.display = 'none';
        });
    });

    // Prevent form resubmission on refresh
    if (window.history.replaceState) {
        window.history.replaceState(null, null, window.location.href);
    }
</script>
```

Figure 16: Snippet code of the error login function

Registration is the function that allows the user to create a unique username and password for their account.

Here is the pseudocode of the registration function.

Start session

If form is submitted (POST request):

    Get username, password, and email from the form

    If all fields are not empty:

        Generate a random user ID

        Insert new user data into the database

        Redirect to the login page

    Else:

        Show error message

Display signup form:

    - Input fields for username, password, and email

    - Submit button

    - Link to login page for existing users

```php
<?php
session_start();

    include('function.php');
    include('connect.php');

    if($_SERVER['REQUEST_METHOD'] == "POST")
    {
        //something was posted
        $username = $_POST['username'];
        $password = $_POST['password'];
        $email = $_POST['email'];

        if(!empty($username) && !empty($password) && !empty($email))
        {

            //save to database
            $user_id = random_num(20);
            $query = "insert into registration (user_id,username,password,email) values ('$user_id','$username','$

            mysqli_query($con, $query);

            header("Location: login.php");
            die;
        }else{
            echo "Please enter some valid information!";
        }
    }
?>
```

Figure 17: Snippet code of the registration function

The registration function begins by checking the user input in the textbox to ensure the input type is correct. If the input type is incorrect, an error message will be displayed to notify the user to correct it. Once the input is validated, the function will collect the data, such as the username, password, and email, and send it to the database for storage. Each user will be assigned a unique ID number. If the registration process is successful, the page will redirect the user to the login page so they can access their account.

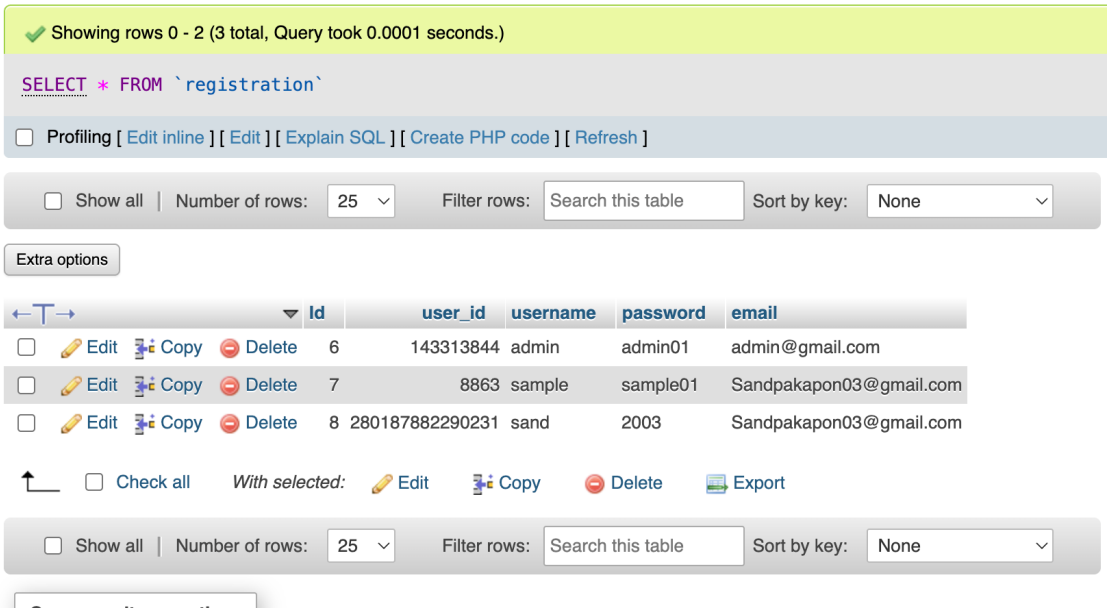

Figure 18: The illustration of how the database is stored after the user has created an account

Interactive quizzes and test features

The purpose of this functionality is to enhance user engagement by providing interactive quizzes and tests that promote active learning. These features are designed to offer users a more immersive and hands-on educational experience, enabling them to reinforce their understanding of the content presented on the website. By incorporating interactivity, learners are more likely to stay motivated and retain information effectively.

Most of the functions will use JavaScript to create and adapt, making them more efficient. In this project, I have three types of quizzes for English vocabulary, three types for English grammar, and also for mathematics.

The example that I will use to explain it will be the test functions.

English vocabulary quiz

```
<script>
    const animals = [
        { name: 'Lion', image: 'animal/lion.jpg' },
        { name: 'Tiger', image: 'animal/tiger.jpg' },
        { name: 'Elephant', image: 'animal/elephant.jpg' },
        { name: 'Giraffe', image: 'animal/giraffe.jpg' },
        { name: 'Butterfly', image: 'animal/butterfly.jpg' },
        { name: 'Cat', image: 'animal/cat.jpg' },
        { name: 'Dog', image: 'animal/dog.jpg' },
        { name: 'Dolphin', image: 'animal/dolphin.jpg' },
        { name: 'Duck', image: 'animal/duck.jpg' },
        { name: 'Fox', image: 'animal/fox.jpg' },
        { name: 'Hamster', image: 'animal/hamster.jpg' },
        { name: 'Horse', image: 'animal/horse.jpg' },
        { name: 'Koala', image: 'animal/koala.jpg' },
        { name: 'Panda', image: 'animal/panda.jpg' },
        { name: 'Polar Bear', image: 'animal/polarbear.jpg' },
        { name: 'Rabbit', image: 'animal/rabbit.jpg' },
        { name: 'Rhino', image: 'animal/rhino.jpg' },
        { name: 'Sheep', image: 'animal/sheep.jpg' },
        { name: 'Turtle', image: 'animal/turtle.jpg' },
        { name: 'Pig', image: 'animal/pig.jpg' },
    ];
```

Figure 19: Example array of English quizzes

The array of English vocabulary will contain the name of the subject and the link that matches the subject. Figure 19 is an example of the animal vocabulary, which will contain animal names with the image link.

The pseudocode illustrates that once the user enters the number of pages, the timer function will be activated to display the time and begin counting down. At the same time, a question will be presented, randomly selected from three different types.

```
function generateQuestion() {
    if (currentQuestionIndex >= 10) {
        clearInterval(timerInterval);
        displayAnswers();
        return;
    }

    const animal = shuffledAnimals[currentQuestionIndex];
    const questionType = Math.floor(Math.random() * 3);


    document.getElementById('animalImage').src = '';
    document.getElementById('resultMessage').innerText = '';
    document.getElementById('answers').innerHTML = '';
    document.getElementById('inputContainer').style.display = 'none';
    document.getElementById('imageContainer').style.display = 'none';
    document.getElementById('userInput').value = '';

    document.getElementById('questionCounter').innerText = `Question: ${currentQuestionIndex + 1}/10`;

    switch (questionType) {
        case 0: // Animal Name Quiz
            document.getElementById('animalImage').src = animal.image;
            document.getElementById('question').innerText = 'What animal is this? Type your answer:';
            document.getElementById('inputContainer').style.display = 'block';
            results[currentQuestionIndex] = { question: `Q${currentQuestionIndex + 1}`, correct: null };
            break;
```

Figure 20: Generating the question function 1

```
case 1: // Animal Image Quiz (select the correct image)
    document.getElementById('question').innerText = `Which image is ${animal.name}?`;
    const images = [animal.image, ...getWrongImages(animal)];
    shuffleArray(images);
    const imageContainer = document.getElementById('imageContainer');
    imageContainer.innerHTML = '';
    images.forEach(image => {
        const imgElement = document.createElement('img');
        imgElement.src = image;
        imgElement.alt = animal.name;
        imgElement.className = 'answer-img';
        imgElement.onclick = () => checkImageAnswer(image, animal.image);
        imageContainer.appendChild(imgElement);
    });
    imageContainer.style.display = 'block';
    results[currentQuestionIndex] = { question: `Q${currentQuestionIndex + 1}`, correct: null };
    break;

case 2: // Animal Identification Quiz
    const answers = [animal.name, ...getWrongNames(animal)];
    shuffleArray(answers);
    document.getElementById('question').innerText = 'What animal is this?';
    const answersContainer = document.getElementById('answers');
    answersContainer.innerHTML = '';
    answers.forEach(answer => {
        const button = document.createElement('button');
        button.innerText = answer;
        button.className = 'answer-btn';
        button.onclick = () => checkIdentificationAnswer(answer === animal.name);
        answersContainer.appendChild(button);
    });
    document.getElementById('animalImage').src = animal.image;
    results[currentQuestionIndex] = { question: `Q${currentQuestionIndex + 1}`, correct: null };
    break;
}
```

Figure 21: Generating the question function 2

As Figures 20 and 21 show, the generateQuestion() function is a central component of the interactive quiz system and is designed to dynamically present one of ten quiz questions to the user. It first performs a boundary check by evaluating whether the current question index has reached ten, signifying the end of the quiz session. If this condition is met, the function terminates the countdown timer and calls the displayAnswers() function to present the final results to the user. If the quiz is still in progress, the function retrieves the current animal object from the shuffled list and selects one of three predefined question types through randomisation, using Math.floor(Math.random() * 3).

Before displaying a new question, the function systematically resets the interface by clearing the animal image, any displayed result messages, previously shown answers, and input values. It also updates the question counter to reflect the user's progression through the quiz.

The function then utilises a switch statement to determine which type of question to generate:

- **Case 0 (Animal Name Quiz):** The system displays an image of the animal and prompts the user to type its name. This format evaluates the learner's ability to recall and spell the animal's name correctly.
- **Case 1 (Image Selection Quiz):** The name of an animal is shown as the prompt, and the user is required to select the correct image from a group of three, which includes the correct image and two distractors generated by the helper function getWrongImages(). The image options are shuffled to prevent pattern recognition.
- **Case 2 (Multiple Choice Identification):** An animal image is displayed, and the user must choose the correct animal's name from three text-based options. These names are generated by combining the correct answer with two incorrect alternatives provided by getWrongNames() and are also shuffled for randomness.

Each question instance is recorded in the results array, with the outcome initially set to null to indicate that it has not yet been evaluated. Upon receiving a user response, one of the checking functions—checkImageAnswer(), checkIdentificationAnswer(), or checkTextAnswer()—is invoked based on the question type. These functions determine whether the response is correct, update the result accordingly, and proceed to either the next question or the final result display, depending on the current index.

After the user has answered all the questions, or if the timer finishes before the user completes them, the `displayAnswers()` function is executed. This function compiles the user's performance across all questions, displaying whether each response was correct or incorrect. It also provides a summary of the total number of correct answers out of ten.

To support the quiz logic, the functions `getWrongImages()` and `getWrongNames()` provide distractor options by filtering out the correct animal and randomly selecting two incorrect alternatives. The `shuffleArray()`

function ensures that the answer options are presented in a random order, enhancing the variability and fairness of the quiz experience.

Finally, the quiz is initialised within the `window.onload` event handler, which triggers three key operations: shuffling the quiz questions, starting a countdown timer set to 120 seconds, and generating the first question. This comprehensive approach ensures that each quiz session is both unique and pedagogically effective.

English grammar quiz

There are three types of quizzes: multiple choice, fill-in-the-blank, and true or false.

```
function loadQuiz() {
    const quizContainer = document.getElementById('quiz');
    questions.forEach((q, index) => {
        const questionElement = document.createElement('div');
        questionElement.innerHTML = `<h3 class="question">${q.question}</h3>`;
        q.choices.forEach(choice => {
            questionElement.innerHTML += `
                <label>
                    <input type="radio" name="question${index}" value="${choice}">
                    ${choice}
                </label><br>
            `;
        });
        quizContainer.appendChild(questionElement);
    });
}

function submitQuiz() {
    let score = 0;
    questions.forEach((q, index) => {
        const selected = document.querySelector(`input[name="question${index}"]:checked`);
        if (selected && selected.value === q.answer) {
            score++;
        }
    });
    document.getElementById('result').innerText = `You scored ${score} out of ${questions.length}`;
}

document.getElementById('submit').addEventListener('click', submitQuiz);

window.onload = loadQuiz;
```

Figure 22: Snipped code of the multipole choice

The provided code snippet (Figure 19) implements a simple multiple-choice quiz with two main functions: `loadQuiz()` and `submitQuiz()`. These functions work together to dynamically display quiz content, capture user responses, and calculate the final score.

The `loadQuiz()` function generates the quiz questions and answer choices within the HTML document. It accesses the container element with the ID `quiz` and iterates over the `questions` array using `forEach()`. For each question, it creates a `<div>` with an `<h3>` heading for the question text and appends a `<label>` with a radio button (`<input type="radio">`) for each answer choice. This structure allows users to select only one answer per question, maintaining

the quiz's format. Once all choices are added, the completed question is displayed in the `quizContainer`.

When the user clicks the submit button, the `submitQuiz()` function is triggered. This function evaluates the selected answers and calculates the score. It initialises a score counter, checks each question for the selected answer, and increments the score if the answer is correct. After processing all questions, it updates the element with the ID `result` to show the total number of correct answers.

An event listener is attached to the submit button using `addEventListener()`, which connects the `submitQuiz()` function to the button click. This ensures the answers are evaluated only after the quiz is submitted.

Finally, the quiz interface is rendered on page load using `window.onload` to call the `loadQuiz()` function. This approach allows for dynamic content generation based on the question data structure.

```
function loadQuiz() {
    const quizContainer = document.getElementById('quiz');
    questions.forEach((q, index) => {
        const questionElement = document.createElement('div');
        questionElement.classList.add('question');
        questionElement.innerHTML = `<h3 class="question">${q.question}</h3>
            <input type="text" id="answer${index}" placeholder="Type your answer here">`;
        quizContainer.appendChild(questionElement);
    });
}

function submitQuiz() {
    let score = 0;
    questions.forEach((q, index) => {
        const userAnswer = document.getElementById(`answer${index}`).value.trim().toLowerCase();
        if (userAnswer === q.answer.toLowerCase()) {
            score++;
        }
    });
    document.getElementById('result').innerText = `You scored ${score} out of ${questions.length}`;
}

document.getElementById('submit').addEventListener('click', submitQuiz);

window.onload = loadQuiz;
```

Figure 23: Snipped code of the fill-in-the-blank

The code snippet (Figure 20) implements a text-based quiz system using dynamic content generation and user input evaluation. It features two main functions: `loadQuiz()` and `submitQuiz()`, which render quiz questions and assess user responses, respectively. This design adheres to web development principles by interacting with the Document Object Model (DOM) for enhanced interactivity.

The `loadQuiz()` function runs when the web page loads and dynamically generates the quiz interface from a predefined array of question objects. Each question object contains a text prompt (`q.question`) and the correct answer (`q.answer`). The function first obtains a reference to the `quiz` container and

iterates through the questions using `forEach()`. For each question, it creates a `div` with a CSS class of `question`, inserts the question text using an `<h3>`, and adds an input element for user responses. Each input field has a unique ID (`answer${index}`) for easy retrieval of answers during submission.

The `submitQuiz()` function is triggered by a click event on the submit button (ID `submit`). It calculates the score by initialising a counter and incrementing it for each correct answer. The function loops through the questions, retrieves user inputs, and compares them to the correct answers in a case-insensitive manner. The final score is displayed in the HTML element with ID `result`, indicating the number of correct answers out of the total.

This JavaScript-based approach provides a flexible and user-friendly quiz application, fostering active engagement and knowledge reinforcement. It is scalable, allowing educators or developers to easily modify questions or functionality.

```
function loadQuiz() {
    const quizContainer = document.getElementById('quiz');
    questions.forEach((q, index) => {
        const questionElement = document.createElement('div');
        questionElement.classList.add('question');
        questionElement.innerHTML = `
            <h3 class="question">${q.question}</h3>
            <label>
                <input type="radio" name="question${index}" value="True"> True
            </label>
            <label>
                <input type="radio" name="question${index}" value="False"> False
            </label>
        `;
        quizContainer.appendChild(questionElement);
    });
}

function submitQuiz() {
    let score = 0;
    questions.forEach((q, index) => {
        const selected = document.querySelector(`input[name="question${index}"]:checked`);
        if (selected && selected.value === q.answer) {
            score++;
        }
    });
    document.getElementById('result').innerText = `You scored ${score} out of ${questions.length}`;
}
chat, ⌘K to generate
document.getElementById('submit').addEventListener('click', submitQuiz);

window.onload = loadQuiz;
```

Figure 24: Snipped code of the true or false

The code snippet (Figure 24) demonstrates a True or False quiz system using client-side scripting and dynamic DOM manipulation. It consists of two main functions: loadQuiz() and submitQuiz(), which generate the quiz interface and evaluate user responses, respectively.

The loadQuiz() function renders quiz questions on the web page when the window loads. It retrieves an HTML element with the ID quiz to serve as the

container, then iterates over an array of question objects. For each question, it creates a new element with the class name question, inserts the question text in an <h3> tag, and adds two radio button inputs labelled “True” and “False” grouped by name to allow for single selections.

The submitQuiz() function is triggered when the submit button is clicked. It initialises a score variable and evaluates user responses by checking which radio button is selected for each question. If the selected answer matches the correct one, the score is incremented. After evaluation, the final score is displayed in an element with the ID result, providing feedback on performance.

Overall, this implementation effectively creates an interactive True or False quiz using vanilla JavaScript, supporting real-time user interaction and immediate feedback, which enhances learning and knowledge retention.

Mathematic

```
function generateQuestion() {
    if (currentQuestion >= totalQuestions) {
        showResults();
        return;
    }

    const quizType = Math.floor(Math.random() * 3);
    if (quizType === 0) {

        num1 = Math.floor(Math.random() * 1000) + 1;
        num2 = Math.floor(Math.random() * 1000) + 1;
        correctAnswer = num1 + num2;
        document.getElementById('question').innerText = `What is ${num1} + ${num2}?`;
    } else if (quizType === 1) {

        num1 = Math.floor(Math.random() * 100) + 1;
        num2 = Math.floor(Math.random() * 100) + 1;
        correctAnswer = num1 + num2;
        document.getElementById('question').innerText = `What is ${num1} + ${num2}?`;
    } else {

        const images = [
            { value: 1, src: 'mathematics/1.jpg' },
            { value: 2, src: 'mathematics/2.jpg' },
            { value: 3, src: 'mathematics/3.jpg' },
            { value: 4, src: 'mathematics/4.jpg' },
            { value: 5, src: 'mathematics/5.jpg' },
            { value: 6, src: 'mathematics/6.jpg' },
            { value: 7, src: 'mathematics/7.jpg' },
            { value: 8, src: 'mathematics/8.jpg' },
            { value: 9, src: 'mathematics/9.jpg' },
            { value: 10, src: 'mathematics/10.jpg' }
        ];
        const img1Index = Math.floor(Math.random() * images.length);
        const img2Index = Math.floor(Math.random() * images.length);
        correctAnswer = images[img1Index].value + images[img2Index].value;
        document.getElementById('question').innerHTML = `What is the sum of the following images? <img src=
```

Figure 25: Snipped code of the mathematical test function 1

The snipped code (Figures 25 and 26) implements an interactive mathematics quiz designed for educational purposes, focusing on various types of arithmetic questions to enhance engagement and assess basic numeracy skills. It utilises logical conditionals, DOM manipulation, and interval timing to create a timed quiz with twenty randomly generated questions, supported by key functions such as startTimer(), generateQuestion(), checkAnswer(), updateQuestionCount(), and showResults().

The quiz activates when the web page loads, starting the timer at 180 seconds, which displays the countdown in “MM:SS” format. This feature helps learners

manage their time. If the timer runs out before all questions are completed, the quiz ends and triggers the showResults() function.

The generateQuestion() function creates three types of arithmetic problems: (1) addition of large numbers (1 to 1000), (2) addition of smaller numbers (1 to 100), and (3) image-based addition with visuals depicting numbers from 1 to 10. This approach supports various learning styles, especially for visual learners. Each generated question increments the current question counter, updating the progress for the learner.

```
function updateQuestionCount() {
    document.getElementById('questionCount').innerText = `Questions: ${currentQuestion}/${totalQuestions}`;
}


function checkAnswer() {
    const userAnswer = document.getElementById('answer').value;

    if (userAnswer === '' || isNaN(userAnswer)) {
        return;
    }

    const answer = parseInt(userAnswer);
    results.push({ question: document.getElementById('question').innerText, userAnswer: `<span style="color: bl

    if (answer === correctAnswer) {
        score++;
    }


    document.getElementById('answer').value = '';
    generateQuestion();
}

// Function to show results at the end of the quiz
function showResults() {
    clearInterval(timer);
    const resultElement = document.getElementById('result');
    resultElement.innerHTML = `Quiz Over! You scored ${score} out of ${totalQuestions}.<br><br>Results:<br>`;
    results.forEach((result, index) => {
        resultElement.innerHTML += `Q${index + 1}: ${result.question} <span style="color: blue;"> Your answer:
    });
}
```

Figure 26: Snipped code of the mathematical test function 2

User input is processed through the checkAnswer() function, which validates responses and compares them with the correct answers. Both the learner's and correct answers are stored for feedback, and the score is updated if the answer is correct. After answering all questions, the showResults() function presents the final score, showing each question alongside the learner's answer (in blue) and the correct answer (in red). This provides immediate feedback, helping students recognise and understand their mistakes.

Overall, the code exemplifies a digital learning tool that integrates formative assessment, visual strategies, and user feedback, reflecting sound pedagogical practices and promoting active engagement in mathematics.

Tutor booking

```php
<?php
session_start();
    include('function.php');
    include('connect.php');

    if(isset($_POST['save_datetime']))
    {
        $name = $_POST['name'];
        $datetime = $_POST['datetime'];


        $checkQuery = "SELECT * FROM booking WHERE name='$name' AND datetime='$datetime'";
        $checkResult = mysqli_query($con, $checkQuery);

        if (mysqli_num_rows($checkResult) > 0) {

            $_SESSION['status'] = "Error: This time has already been booked, Please try booking diffrence time";
        } else {

            $query = "INSERT INTO booking (name,datetime) VALUES ('$name','$datetime')";

            $query = mysqli_query($con, $query);
 ⌘L to chat, ⌘K to generate
            if($query)
            {
                $_SESSION['status'] = "You have booked the tutor.";
                header("Location: payment.html");
                exit();
            }
            else{
                $_SESSION['status'] = "Error: Could not complete the booking.";
            }
        }
    }

?>
```

Figure 27: Snipped code of the tutor booking

The PHP and HTML script described here functions as an interactive tutoring session booking interface on an educational platform. It enables users to schedule appointments with qualified tutors by integrating client-side and server-side elements for a seamless experience, while ensuring proper data validation and time-slot management through a PHP backend and Mysql database.

When the booking form is submitted, the script initiates a PHP session with `session_start()`. It includes external files for database connection (`connect.php`) and additional functions (`function.php`). Upon clicking the “Booking” button, the script captures the selected tutor’s name, date, and time from the form. To prevent double-booking, it runs a SELECT query to see if the tutor is already booked during that time. If the slot is unavailable, a status message informs the user.

If the time is available, the script inserts a new booking record into the database. Upon successful execution, the user is redirected to a `payment.html` page to finalise the booking with payment. If the query fails, an error message is displayed.

The front end includes a navigation bar linking to different sections of the platform, a disclaimer about the paid service, and a scrollable table of available tutors and their qualifications. A form below the table allows users to select a tutor and choose a date and time, enforcing proper input formatting.

```php
<?php
// Display status message
if (isset($_SESSION['status'])) {
    echo "<div class='status-message'>" . $_SESSION['status'] . "</div>";
    unset($_SESSION['status']);
}
?>
```

Figure 28: Snipped code of the tutor booking

Session-based feedback messages are dynamically displayed below the form to inform users of the status of their booking attempts. The script concludes with a button linking back to the home page.

In summary, this booking system provides a functional approach to interactive scheduling in an e-learning environment, integrating form handling, validation, feedback mechanisms, and database interaction for a user-friendly experience.

Contracting

This is the pseudocode of the contracting page;

Start HTML page

Set page title and include styles

Display navigation bar

Show heading: "Contact Form"

Create a form:

- Set form action to Web3Forms API

- Set method to POST

Add a hidden input for access key

Add text input for the user's name (required)

Add text input for the user's email (required)

Add a textarea for the user's message (required)

Add a "Send" button to submit the form

End HTML page

The provided HTML code outlines the structure and functionality of a Support Contact Form embedded within an educational platform called “Brainwave.” This form serves as a straightforward channel for users to reach out to the platform’s support team for inquiries, feedback, or any issues encountered while using the system. The design features a navigation bar that facilitates seamless access to essential sections of the site, including Home, English, Mathematics, Account, and Support, thereby enhancing the overall user experience.

At the core of the page is the contact form, which leverages a third-party service, Web3Forms, to manage form submissions. This is indicated by the form’s action attribute, directing input data to the Web3Forms API endpoint. An access key, included as a hidden input field, authorises the form to interact with the Web3Forms platform. The form captures three primary pieces of user input: the sender’s name, email address, and their message. These fields are marked as required, ensuring that the form cannot be submitted unless all necessary information is provided.

The styling of the page is handled externally through two CSS files: support.css for the specific styling of the form and navbar.css to maintain consistent aesthetics for the navigation bar across the website. The submission button, styled with the class name submit-btn, triggers a POST request upon user interaction. This approach abstracts backend processing of messages, reducing server-side complexity and providing a user-friendly communication channel for support-related inquiries. Overall, this implementation reflects best practices in web design, including responsiveness, form validation, and integration with external services.

Account

```php
// Check if user_id is set in the session
if (!isset($_SESSION['user_id'])) {
    header("Location: login.php");
    exit();
}

if ($_SERVER['REQUEST_METHOD'] == "POST" && isset($_POST['update'])) {
    // Get the user ID from the session
    $userId = $_SESSION['user_id'];

    // Get the posted data
    $username = $_POST['username'];
    $password = $_POST['password'];
    $email = $_POST['email'];

    if (!empty($username) && !empty($email)) {
        // Prepare the update query
        $query = "UPDATE registration SET username = ?, email = ?" . (!empty($password) ? ", password = ?" : ""

        // Prepare the statement
        $stmt = mysqli_prepare($con, $query);

        if (!empty($password)) {
            mysqli_stmt_bind_param($stmt, "sssi", $username, $email, $password, $userId);
        } else {
            mysqli_stmt_bind_param($stmt, "ssi", $username, $email, $userId);
        }

        // Execute the statement
        if (mysqli_stmt_execute($stmt)) {
            echo "<script>alert('User details updated successfully!');</script>";
        } else {
            echo "<script>alert('Failed to update user details.');</script>";
        }
    } else {
        echo "Please enter some valid information!";
    }
}
```

Figure 29: Snipped code updating account

The provided PHP script facilitates user account management within a web-based application. It primarily allows authenticated users to update their personal information, including their username, email, and optionally, their password, or to permanently delete their account. The script begins by initiating a session with `session_start()` and includes essential external files (`function.php` and `connect.php`) to support its functionalities and establish a database connection.

A conditional check is performed to ensure that a valid session exists; specifically, it verifies that the `user_id` is set. If not, the user is redirected to the login page to prevent unauthorised access.

When the form is submitted via a POST request and the "update" action is detected, the script retrieves the submitted data (username, email, and password) along with the user's ID from the session. It conducts a validation check to ensure that the necessary fields are not empty. If the password field is filled in, the SQL UPDATE query is constructed to include the password; otherwise, only the username and email are updated. This query is executed

securely using prepared statements to mitigate the risk of SQL injection attacks. Following the successful execution of the query, a JavaScript alert notifies the user of the successful update; if the execution fails, a failure message is displayed.

```php
// Handle account deletion
if ($_SERVER['REQUEST_METHOD'] == "POST" && isset($_POST['delete_account'])) {
    $userId = $_SESSION['user_id'];

    // Prepare the delete query
    $deleteQuery = "DELETE FROM registration WHERE user_id = ?";
    $deleteStmt = mysqli_prepare($con, $deleteQuery);
    mysqli_stmt_bind_param($deleteStmt, "s", $userId);

    // Execute the delete statement
    if (mysqli_stmt_execute($deleteStmt)) {
        // Log out the user and redirect to the home page
        session_destroy();
        header("Location: index.php");
        exit();
    } else {
        echo "<script>alert('Failed to delete account.');</script>";
    }
}
```

Figure 30: Snipped code deletes the user account

The script also includes functionality for account deletion. When the "delete_account" POST action is triggered, it prepares and executes a SQL DELETE query using the user's ID. Upon successful deletion, the session is terminated, and the user is redirected to the homepage, effectively logging them out and removing their data from the database.

The HTML structure features a form-based user interface. It retrieves the logged-in user's data using a prepared SELECT query and dynamically populates the form fields with the existing data using PHP echo statements. To enhance security and prevent XSS (Cross-Site Scripting) attacks, the data is sanitised with the htmlspecialchars() function. The form includes input fields for the username, email, and an optional password, along with buttons to submit updates or delete the account.

A navigation bar provides links to other sections of the platform, while user interaction elements such as a logout button and a back-to-home button improve usability. Overall, the script adheres to core principles of secure web development, including session management, prepared statements, and user feedback, creating a robust user account management interface.

7. Testing

Feature	Test case ID	Test Description	Test step	Expected result	Status
Registration Function	01	Test if user can register with valid data	1. Navigate to registration page. 2. Enter valid username, email, and password. 3. Submit the form.	User is registered, redirected to the login page.	Yes
	02	Test if registration fails with missing data	1. Navigate to registration page. 2. Leave one or more fields empty. 3. Submit the form.	Error message displayed, user remains on the registration page.	Yes
	17	Test if registration fails with invalid email format	1. Navigate to registration page. 2. Enter an invalid email (e.g., no “@” symbol). 3. Submit the form.	Error message displayed: “Invalid email format”.	Yes
Login Function	03	Test if user can log in with valid credentials	1. Navigate to login page. 2. Enter valid username and password. 3. Submit the form.	User is logged in, redirected to home page.	Yes

	04	Test if login fails with invalid credentials	1. Navigate to login page. 2. Enter incorrect username or password. 3. Submit the form.	Error message displayed, user remains on the login page.	Yes
	18	Test if user can reset password through email verification	1. Click on “Forgot Password”. 2. Enter registered email. 3. Check email for reset link.	Password reset link sent, user able to change password.	No
Quiz Function (English Vocabulary)	05	Test if the quiz generates random questions correctly	1. Start the English vocabulary quiz. 2. Verify if questions are random and diverse (animal names, images, etc.).	Randomised questions are displayed.	Yes
	06	Test if the timer works correctly during the quiz	1. Start the English vocabulary quiz. 2. Verify if the countdown timer begins and runs correctly.	Timer begins at 120 seconds, counts down correctly.	Yes
	19	Test if user can review answers after completing the quiz	1. Complete the quiz. 2. Verify that answers are displayed with feedback.	All answers displayed with feedback and correct/incorrect status.	Yes

Quiz Function (Mathematics)	07	Test if mathematics questions are generated randomly and correctly	1. Start the mathematics quiz. 2. Verify if questions are arithmetic-based and diverse.	Random arithmetic questions are generated.	Yes
	08	Test if the math quiz displays the correct answer after user input	1. Answer a math question. 2. Check if the result is correct and if feedback is displayed.	Correct answer feedback shown, score updated.	Yes
	20	Test if math quiz includes images or diagrams when needed	1. Start the math quiz. 2. Verify that questions may contain diagrams or images for questions related to geometry.	Diagrams or images are correctly displayed where appropriate.	Yes
Tutor Booking Function	09	Test if tutor booking slot is available	1. Navigate to tutor booking page. 2. Choose a tutor and time slot. 3. Check availability.	Slot is either available or a message is shown if unavailable.	Yes
	10	Test if booking submission works correctly when a slot is available	1. Select a tutor and time. 2. Submit the booking. 3. Verify if the booking is recorded in the database.	Booking is successfully recorded, user redirected to payment.	Yes

	21	Test if tutor booking prevents overlapping sessions	1. Check if the same tutor is booked for overlapping sessions. 2. Try booking an overlapping time.	Error message displayed: “This tutor is already booked for that time”.	Yes
Account Update Function	11	Test if user can update personal information successfully	1. Log in and navigate to account settings. 2. Modify username/email. 3. Submit the form.	Information is updated and saved correctly in the database.	Yes
	12	Test if user can delete account successfully	1. Log in and navigate to account settings. 2. Choose to delete account. 3. Confirm deletion.	Account is deleted, user logged out and redirected to homepage.	Yes
	22	Test if user can change password successfully	1. Log in and navigate to account settings. 2. Change password. 3. Submit changes.	Password is updated, user logged out and redirected to login page.	Yes
Contact Form	13	Test if the contact form sends data correctly	1. Navigate to contact form. 2. Enter name, email, and message.	Form data is submitted and received by Web3Forms API.	Yes

			3. Submit the form.		
	23	Test if contact form shows an error for invalid email address	1. Enter an invalid email address in the contact form (e.g., “user.com”). 2. Submit the form.	Error message displayed: “Invalid email address”.	Yes
Session Management	14	Test if session starts correctly for logged-in user	1. Log in with valid credentials. 2. Navigate to any protected page.	Session should be active, user should be able to access the page.	Yes
Error Handling	15	Test if error message appears when a field is left empty in the login/registration form	1. Leave a field empty in the login/registration form. 2. Submit the form.	Error message displayed, user remains on the page.	Yes
	24	Test if server error is handled gracefully (e.g., 500 Internal Server Error)	1. Simulate a server error or API failure. 2. Perform any action on the platform.	Error page shown, with message: “Something went wrong. Please try again later.”	Yes
Security	16	Test if SQL injection is prevented in login and registration forms	1. Enter SQL code (e.g., '; DROP TABLE users;--) into the login/registration form. 2. Submit the form.	Form submission is blocked or handled securely.	Yes
	25	Test if XSS attacks are prevented in	1. Enter a script tag <script>alert('XSS')</script>	Script is sanitized, no alert or malicious	Yes

		user input fields	in the registration/log in form. 2. Submit the form.	script executed.	

Table 4. Testing table

7.1 Test coverage

Software testing is an indispensable component in the development lifecycle, serving to assure that a system meets both its functional and non-functional requirements. Within the context of designing an interactive e-learning platform for primary-level students, two predominant testing methodologies- black box testing and white box testing- prove instrumental for the verification and validation of the system. The applicability of each approach is contingent upon the specific nature of the requirements being assessed.

Black Box Testing

Black box testing is fundamentally concerned with assessing the outputs of a system based on given inputs, without regard for the internal mechanisms of the code. This methodology is particularly effective for confirming that the system functions as anticipated from the user's perspective.

For instance, in relation to the requirement R1: “The system shall allow users to register with a valid email and password,” black box testing facilitates the simulation of user interactions to ascertain the accuracy of the registration process. Test Case TC01, for example, evaluates whether valid input yields successful registration outcomes. In contrast, TC02 and TC17 are designed to examine the system's response to scenarios of missing or invalid data. The internal workings of the registration function are extraneous to this evaluation; the focal point is the system's ability to accept or reject inputs appropriately.

Furthermore, considering requirement R4: “The system shall provide a quiz interface for English vocabulary utilising images and words,” black box testing (as illustrated by TC05 and TC06) assesses the quiz's ability to randomly present questions while accurately recording responses within a stipulated 120-second timeframe. These evaluations establish whether the quiz functionality aligns with the specified requirement without delving into the underlying algorithms or data structures.

The tutor booking functionality (R6) also gains from black box testing. Test Cases TC09, TC10, and TC21 scrutinise whether the booking system

accurately displays available slots, processes user selections, and mitigates the possibility of overlapping bookings. Such assessments validate the observable behaviour of the booking system against anticipated outcomes.

In summation, black box testing is pivotal in ensuring that the e-learning platform delivers a consistent, user-friendly experience, particularly across critical areas such as user registration, login processes, quizzes, tutor bookings, and form submissions.

White Box Testing

Conversely, white box testing entails a detailed examination of the internal logic, control flow, and data paths within the code. This approach is particularly valuable for verifying the correct function of each component, including loops and conditional branches.

Taking requirement R3: “The system shall validate login credentials securely” as an example, white box testing is essential for scrutinising the authentication logic. Test Case TC16 focuses on SQL injection vulnerabilities, a threat that can only be adequately addressed by analysing how input data is processed and sanitised before interfacing with the database. Similarly, TC26 evaluates the application’s resistance to cross-site scripting (XSS) by investigating the mechanisms employed to encode or filter user inputs.

Moreover, white box testing is critical for R7: “The system shall maintain user sessions and expire them after periods of inactivity.” While a user-facing test (TC24) can verify session expiration, only white box testing can comprehensively validate the implementation of session timeout logic and ascertain its activation under conditions of user inactivity.

Within the realm of quizzes, Test Cases TC19 and TC20 can be further examined through white box testing to ensure the correct functionality of scoring algorithms, the secure implementation of randomisation, and the correct retrieval of image assets from designated directories.

White box testing is also instrumental in assessing performance and efficiency. Test cases relating to responsive design (such as TC27 and TC28) may benefit from a white box review to confirm the proper adaptation of the interface across various screen sizes through CSS media queries and JavaScript functions.

In conclusion, both black box and white box testing approaches are indispensable for ensuring the quality and reliability of the e-learning platform. While black box testing emphasises user expectations and functional requisites without probing into the code, white box testing is essential for verifying the integrity and security of the underlying implementation.

The integration of both methodologies—utilising black box testing for high-level functionality and user interactions, alongside white box testing for internal logic and security—ensures a comprehensive validation process. Test cases such as TC01, TC05, and TC10 exemplify effective black box testing, whereas TC16, TC24, and TC26 demonstrate the depth and assurance that white box testing contributes to the development process.

7.2 Test methodology

Rigorous evaluation of system outputs is essential for developing educational technologies that provide meaningful and user-centric learning experiences. This section outlines the methods used to test the output of the interactive e-learning platform designed for primary-level learners and describes the process of gathering and incorporating feedback from both expert and non-expert users.

Output Testing Procedures

The output testing phase focused on ensuring that the platform delivered correct visual, textual, and interactive responses to user inputs across core functionalities such as registration, login, quizzes, and tutor bookings. This approach combined functional testing, usability evaluation, and interface responsiveness analysis, primarily utilising black box testing techniques to assess the platform from the end-user perspective.

In line with Requirement R4, the vocabulary quiz module was tested to confirm the accuracy and clarity of its output. For example, Test Case TC05 involved the user selecting the correct answer from a multiple-choice vocabulary question. The expected output was the display of a “Correct!” message, accompanied by an audio cue and a visual indicator (e.g., a checkmark). If the user selected an incorrect response (TC06), the system was expected to provide corrective feedback and prompt the user to try again. These outputs were verified by observing the interface behaviour and user responses, in accordance with black box testing methodologies.

The login and registration features (Requirements R1 and R3) were evaluated using test cases TC01 and TC02, which examined whether appropriate confirmation or error messages were presented in response to valid and invalid user credentials. The expected outputs included alert dialogues for invalid email formats and confirmation messages upon successful registration. These outputs ensured that the system provided real-time, actionable feedback to guide user interactions.

To assess the system’s compatibility across devices (Requirement R9), Test Case TC28 was conducted on various screen sizes, including smartphones and tablets. The outputs were expected to remain visually coherent, with consistent placement of text, buttons, and navigation elements. This testing ensured that the outputs maintained pedagogical effectiveness, regardless of the user’s device, thereby supporting inclusive learning.

The contact and feedback form (Requirement R8) also underwent output testing. Test Cases TC13 and TC14 verified that successful submissions triggered visible confirmation messages and that all fields displayed appropriate validation prompts when left blank. These tests ensured that users understood the system's state and the success of their form completion.

Feedback Collection and Application

Feedback was gathered from both expert users (educators and designers) and non-expert users (students and parents) to enhance the platform's educational value and overall user experience. This feedback informed the iterative development and refinement of the platform’s outputs.

Expert users included two primary school teachers and one user interface designer. The educators assessed how well the output messages and visual cues aligned with age-appropriate learning strategies. One teacher suggested increasing the use of immediate, multimodal feedback, such as audio cues and colour changes, to reinforce correct answers in quizzes. As a result, the quiz output was modified to include animated icons and auditory reinforcement for both correct and incorrect responses.

The user interface designer provided insights on layout and interaction design, noting that some outputs lacked sufficient contrast and clear visual hierarchy. This feedback led to enhancements in the visual presentation of output messages, including increased font size and consistent colour schemes for success and error messages across modules.

Non-expert users, including six children aged 7 to 10 and two parents, participated in guided user testing sessions. The children interacted with the quiz and tutor booking modules while their responses were observed. Some students found the tutor booking process unclear. Consequently, visual aids and instructional tooltips were incorporated to improve the clarity of outputs in that module. Parental feedback emphasised the importance of form error messages and submission confirmations, prompting further refinements in the design of form outputs.

Conclusion

In summary, output testing played a critical role in validating the effectiveness, clarity, and accessibility of the e-learning platform’s interactive features. By combining systematic test cases with comprehensive user feedback, the platform was iteratively refined to ensure that its outputs were not only functionally accurate but also pedagogically effective and user-friendly. The engagement of both expert and non-expert users in the evaluation process contributed to a well-rounded and responsive design, thereby supporting meaningful learning outcomes for primary-level users.

8. Conclusions and reflections

Critical Reflection on the Project Lifecycle of an Interactive E-Learning Platform

The development of an interactive e-learning platform tailored for primary-level learners progressed through a structured project lifecycle, encompassing the stages of initiation, planning, design, implementation, testing, and evaluation. Each phase presented distinct challenges and learning opportunities, culminating in a more profound understanding of the development process and the necessity for effective educational technology. This reflection critically examines each phase, evaluating the effectiveness of the resulting application, the achievement of established objectives, and potential avenues for further enhancement.

Project Initiation and Planning

The initiation phase centred on identifying the necessity for an engaging e-learning platform specifically designed for young learners. A precise articulation of the problem was achieved, emphasising deficiencies in existing online educational tools, particularly their lack of interactivity and the absence of age-appropriate content. The planning stage involved delineating clear objectives, timelines, and deliverables, thereby providing a comprehensive roadmap for the project. Project management tools such as Gantt charts and requirement specifications were instrumental in managing scope and mitigating risks. Despite mostly effective planning, a significant limitation was the underestimation of the time necessary for integrating multimedia elements, which resulted in minor delays in subsequent development stages.

Design and Prototyping

The design phase prioritised user-centric principles, resulting in an interface that is both intuitive and visually appealing for children. Utilising Figma, wireframes and mock-ups were developed to visualise both layout and functionality. Careful consideration was given to colour schemes, button sizes, and content accessibility, ensuring alignment with the cognitive abilities of the target age group. Furthermore, the application of visual learning principles—such as Mayer's Multimedia Learning Theory—fortified the pedagogical underpinnings of the platform. Nonetheless, while the design successfully addressed usability concerns, the limited user testing conducted with actual

students resulted in certain design assumptions not being validated until later stages in development.

Implementation

The implementation phase was executed using technologies including HTML, CSS, JavaScript, and PHP within the Visual Studio Code environment. This phase entailed the translation of the design into a functional web application, incorporating interactive features such as quizzes, animations, and audio effects. The modular structure of the code emerged as a significant strength, enhancing maintainability; however, a notable weakness was identified in the limited responsiveness of the site on mobile devices, underscoring the necessity for more rigorous cross-platform testing. This shortcoming highlighted the importance of adaptive design, particularly in light of the prevalence of tablet and smartphone usage among students.

Testing and Evaluation

The testing phase was conducted through both unit testing and user feedback mechanisms. Test cases were implemented to ensure the functionality of critical components, particularly quizzes and navigation systems. Informal feedback gathered from peers and educators indicated that the application was both engaging and educational. However, a more comprehensive testing regimen involving actual students would likely have yielded more reliable insights. The evaluation affirmed that most core objectives were met, including enhanced interactivity, age-appropriate content, and a user-friendly design. Nevertheless, features such as progress tracking and accessibility support for learners with special needs were not fully implemented due to time constraints.

Research and Findings

The design decisions were guided by extensive academic and market research, which integrated best practices in e-learning and child development. Empirical studies on interactive learning and cognitive load theory influenced the development of multimedia content. The research findings corroborated the significance of combining visual elements, text, and interactivity to enhance engagement and retention among young learners. Additionally, the literature identified a gap in personalised learning experiences for primary students, which this platform sought to partially address via interactive quizzes and feedback mechanisms.

Conclusions and Learning Outcomes

The final application successfully manifested an interactive and visually engaging e-learning environment suitable for primary school learners. While it achieved its core objectives, there remains a clear need for enhancements in terms of responsiveness, accessibility, and personalised learning features. The project illuminated the complex interplay between educational theory, user experience, and technical implementation.

On a personal level, this project constituted a significant learning experience. Newly acquired skills encompass responsive web design, understanding client-server architecture utilising PHP, and fundamental principles of instructional design. Additionally, project management capabilities were enhanced through effective task scheduling, version control, and iterative development processes. A critical lesson derived from this experience is the value of early and frequent user feedback, particularly from the actual target audience.

Future Work

Potential enhancements for future iterations of the platform could include:
- A mobile-first responsive redesign that accommodates a broader range of devices.
- The integration of gamification elements to enhance motivation and engagement.
- Further development of support mechanisms for learners with special educational needs (e.g., text-to-speech capabilities, dyslexia-friendly fonts).

9. References

Behl, A., Jayawardena, N., Pereira, V., Islam, N., Giudice, M.D. and Choudrie, J. (2022). Gamification and e-learning for young learners: A systematic literature review, bibliometric analysis, and future research agenda. *Technological Forecasting and Social Change*, 176, p.121445. doi: https://doi.org/10.1016/j.techfore.2021.121445

Yang, D., Lavonen, J. and Niemi, H. (2018). Online Learning engagement: Critical Factors and Research Evidence from Literature. *Themes in eLearning*, 11(1), pp.1–18. Available at: https://files.eric.ed.gov/fulltext/EJ1204753.pdf.

Kim, S., Song, K., Lockee, B.B. and Burton, J.K. (2018). *Gamification in learning and education: enjoy learning like gaming*. Cham: Springer.

Balalle, H. (2024). Exploring student engagement in technology-based education in relation to gamification, online/distance learning, and other factors: A systematic literature review. *Social Sciences & Humanities Open*, [online] 9, pp.100870-100870.doi:https://doi.org/10.1016/j.ssaho.2024.100870.

Deterding, S., Dixon, D., Khaled, R. and Nacke, L., 2011. From game design elements to gamefulness: defining “gamification”. In: *Proceedings of the 2011 annual conference on Human factors in computing systems*, 2011, pp. 2425-2428.

Mayer, R.E., 2009. *Multimedia learning*. 2nd ed. Cambridge: Cambridge University Press.

Vygotsky, L., 1978. *Mind in society: The development of higher psychological processes*. Cambridge, MA: Harvard University Press.

Anderson, R. (2021). *Web Technologies in Education: A Practical Guide to HTML, CSS, and JavaScript*. TechEd Press.

Jones, M., & Brown, T. (2019). Gamification in digital learning: Engagement and motivation in primary education. *International Journal of Educational Technology*, 14(2), 88–96.

Kumar, S. (2021). The role of development environments in modern web design. *Journal of Software Tools and Applications*, 7(1), 45–52.

Rahman, A., & Gupta, D. (2020). Server-side scripting for interactive e-learning systems: A case study of PHP and MySQL. *International Journal of Web-Based Learning and Teaching Technologies*, 15(4), 60–72.

Smith, L., & Lee, J. (2020). The effectiveness of interactivity in primary-level e-learning platforms. *Journal of E-Learning and Childhood Development*, 12(3), 122–134.

ABCmouse, 2023. *ABCmouse Early Learning Academy*. [online] Available at: https://www.abcmouse.com [Accessed 14 Apr. 2025].

Duolingo, 2022. *Duolingo for Kids*. [online] Available at: https://www.duolingo.com [Accessed 14 Apr. 2025].

eLearning Industry, 2023. *10 Ways Gamification Improves Engagement in E-Learning*. [online] Available at: https://elearningindustry.com [Accessed 14 Apr. 2025].

Khan Academy, 2023. *Khan Academy Kids*. [online] Available at: https://learn.khanacademy.org/khan-academy-kids/ [Accessed 14 Apr. 2025].

Kumar, S., 2021. The role of development environments in modern web design. *Journal of Software Tools and Applications*, 7(1), pp.45–52.

Rahman, A. and Gupta, D., 2020. Server-side scripting for interactive e-learning systems: A case study of PHP and MySQL. *International Journal of Web-Based Learning and Teaching Technologies*, 15(4), pp.60–72.

W3Schools, 2023. *HTML, CSS, and JavaScript Tutorials*. [online] Available at: https://www.w3schools.com [Accessed 14 Apr. 2025].

World Bank, 2020. *The Impact of COVID-19 on Education*. [online] Available at: https://www.worldbank.org [Accessed 14 Apr. 2025].

Anderson, R. (2020) *Security Engineering: A Guide to Building Dependable Distributed Systems*. 3rd edn. Wiley.

BCS (2023) *Code of Conduct*. British Computer Society. Available at: https://www.bcs.org/membership/become-a-member/bcs-code-of-conduct/ [Accessed: 24 April 2025].

Beauchamp, T.L. and Childress, J.F. (2019) *Principles of Biomedical Ethics*. 8th edn. Oxford University Press.

Preist, C., Schien, D. and Blevis, E. (2016) 'Understanding and mitigating the effects of device and cloud service design decisions on the environmental footprint of digital infrastructure', *ACM Conference on Designing Interactive Systems*, pp. 1133–1144.

Selwyn, N. (2016) *Education and Technology: Key Issues and Debates*. 2nd edn. Bloomsbury Publishing.

UNESCO (2021) *Digital Learning and the Digital Divide*. Available at: https://en.unesco.org/themes/ict-education [Accessed: 24 April 2025].

United Nations (2020) *Transforming our World: The 2030 Agenda for Sustainable Development*. Available at: https://sdgs.un.org/2030agenda [Accessed: 24 April 2025].

Voigt, P. and Von dem Bussche, A. (2017) *The EU General Data Protection Regulation (GDPR): A Practical Guide*. Springer.

W3Schools. (2024). *HTML Tutorial*. [online] Available at: https://www.w3schools.com/html/

Mozilla Developer Network. (2024). *HTML, CSS, JavaScript Guides*. [online] https://developer.mozilla.org/

PHP Manual. (2024). *PHP Documentation*. [online] https://www.php.net/manual/en/

10. Bibliography

ABCmouse (2023) *ABCmouse Early Learning Academy*. [online] Available at: https://www.abcmouse.com [Accessed 14 Apr. 2025].

Anderson, R. (2020) *Security Engineering: A Guide to Building Dependable Distributed Systems*. 3rd edn. Wiley.

Anderson, R. (2021) *Web Technologies in Education: A Practical Guide to HTML, CSS, and JavaScript*. TechEd Press.

Balalle, H. (2024) 'Exploring student engagement in technology-based education in relation to gamification, online/distance learning, and other factors: A systematic literature review', *Social Sciences & Humanities Open*, 9, pp.100870–100870. doi: https://doi.org/10.1016/j.ssaho.2024.100870.

BCS (2023) *Code of Conduct*. British Computer Society. Available at: https://www.bcs.org/membership/become-a-member/bcs-code-of-conduct/ [Accessed: 24 April 2025].

Beauchamp, T.L. and Childress, J.F. (2019) *Principles of Biomedical Ethics*. 8th edn. Oxford University Press.

Behl, A., Jayawardena, N., Pereira, V., Islam, N., Giudice, M.D. and Choudrie, J. (2022) 'Gamification and e-learning for young learners: A systematic literature review, bibliometric analysis, and future research agenda', *Technological Forecasting and Social Change*, 176, p.121445. doi: https://doi.org/10.1016/j.techfore.2021.121445.

Deterding, S., Dixon, D., Khaled, R. and Nacke, L. (2011) 'From game design elements to gamefulness: defining "gamification"', *Proceedings of the 2011 Annual Conference on Human Factors in Computing Systems*, pp.2425–2428.

Duolingo (2022) *Duolingo for Kids*. [online] Available at: https://www.duolingo.com [Accessed 14 Apr. 2025].

eLearning Industry (2023) *10 Ways Gamification Improves Engagement in E-Learning*. [online] Available at: https://elearningindustry.com [Accessed 14 Apr. 2025].

Jones, M. and Brown, T. (2019) 'Gamification in digital learning: Engagement and motivation in primary education', *International Journal of Educational Technology*, 14(2), pp.88–96.

Khan Academy (2023) *Khan Academy Kids*. [online] Available at: https://learn.khanacademy.org/khan-academy-kids/ [Accessed 14 Apr. 2025].

Kim, S., Song, K., Lockee, B.B. and Burton, J.K. (2018) *Gamification in Learning and Education: Enjoy Learning Like Gaming*. Cham: Springer.

Kumar, S. (2021) ‘The role of development environments in modern web design’, *Journal of Software Tools and Applications*, 7(1), pp.45–52.

Mayer, R.E. (2009) *Multimedia Learning*. 2nd edn. Cambridge: Cambridge University Press.

Mozilla Developer Network (2024) *HTML, CSS, JavaScript Guides*. [online] Available at: https://developer.mozilla.org/ [Accessed: 14 Apr. 2025].

PHP Manual (2024) *PHP Documentation*. [online] Available at: https://www.php.net/manual/en/ [Accessed: 14 Apr. 2025].

Preist, C., Schien, D. and Blevis, E. (2016) ‘Understanding and mitigating the effects of device and cloud service design decisions on the environmental footprint of digital infrastructure’, *ACM Conference on Designing Interactive Systems*, pp.1133–1144.

Rahman, A. and Gupta, D. (2020) ‘Server-side scripting for interactive e-learning systems: A case study of PHP and MySQL’, *International Journal of Web-Based Learning and Teaching Technologies*, 15(4), pp.60–72.

Selwyn, N. (2016) *Education and Technology: Key Issues and Debates*. 2nd edn. Bloomsbury Publishing.

Smith, L. and Lee, J. (2020) ‘The effectiveness of interactivity in primary-level e-learning platforms’, *Journal of E-Learning and Childhood Development*, 12(3), pp.122–134.

UNESCO (2021) *Digital Learning and the Digital Divide*. [online] Available at: https://en.unesco.org/themes/ict-education [Accessed: 24 April 2025].

United Nations (2020) *Transforming our World: The 2030 Agenda for Sustainable Development*. [online] Available at: https://sdgs.un.org/2030agenda [Accessed: 24 April 2025].

Voigt, P. and Von dem Bussche, A. (2017) *The EU General Data Protection Regulation (GDPR): A Practical Guide*. Springer.

Vygotsky, L.S. (1978) *Mind in Society: The Development of Higher Psychological Processes*. Cambridge, MA: Harvard University Press.

W3Schools (2023) *HTML, CSS, and JavaScript Tutorials*. [online] Available at: https://www.w3schools.com [Accessed: 14 Apr. 2025].

W3Schools (2024) *HTML Tutorial*. [online] Available at: https://www.w3schools.com/html/ [Accessed: 14 Apr. 2025].

World Bank (2020) *The Impact of COVID-19 on Education*. [online] Available at: https://www.worldbank.org [Accessed: 14 Apr. 2025].

Yang, D., Lavonen, J. and Niemi, H. (2018) ‘Online learning engagement: Critical factors and research evidence from literature’, *Themes in eLearning*, 11(1), pp.1–18. Available at: https://files.eric.ed.gov/fulltext/EJ1204753.pdf.

Appendix I

The following video provides a walkthrough of the interactive e-learning platform developed for primary-level learners called Brainwave. It highlights key features such as user registration, interactive quizzes, and gamified learning modules.

Watch the video demonstration here: [https://drive.google.com/file/d/1sUflZ4JaFmStrTJPUviILLogZXDQtAAH/view?usp=sharing]

The website link is here: [https://w1909755.users.ecs.westminster.ac.uk/E-Learning/index.php]

Link to the code (because it is bigger than 50 MB) [https://drive.google.com/file/d/1sWURTUQeYyhbbPmqFzF0FpsV-pvvKXJl/view?usp=drive_link]