FYP Report 2024

Sam Cullen

C00250093

Final Year Project Report

South East Technological University
Carlow

Submission Date: 19/04/2023

Table of Contents

Table of Figures

Acknowledgements

I would like to extend my thanks to the faculty and technical staff of the Computer Science Department at South East Technological University Carlow. Their assistance and resources were instrumental in the successful development and testing of the platform.

Additionally, I appreciate my family and peers for their encouragement and the collaborative environment they provided. Special thanks go to the members of the open-source community whose software and frameworks were helpful components of this project.

Finally, I wish to acknowledge my supervisor, Joseph Kehoe, whose mentorship was crucial in navigating the challenges of this project and providing me with the motivation needed to pursue and complete this endeavour.
Joseph Kehoe’s guidance helped refine my technical skills and his advice was instrumental in overcoming the complex problems I encountered throughout the development of my project

Abstract

This project report details the development of CodeHub, a web-based platform designed to enhance coding skills through targeted practice and instant feedback. Initiated to address the gap in practical coding education, the platform allows users to engage in coding exercises, receive real-time evaluations, and improve problem-solving capabilities in a simulated environment.

Throughout the project, various technical challenges were encountered, including the integration of multiple programming languages and the implementation of a user-friendly interface. These challenges were systematically addressed through innovative solutions that enriched the learning process. The project outcomes demonstrate significant advancements in automated code testing and educational technology, reflecting substantial personal growth and enhanced technical expertise.

As a result, CodeHub has proven to be a valuable educational tool, offering insights into effective coding practice and assessment. This report encapsulates the developmental journey, highlights key learning points, the issues encountered, how they were solved, and suggests directions for future enhancements.

Introduction

The development of software development skills is essential not only for individual advancements but also for contributing effectively to the constantly growing change in the tech industry. With this in mind, I decided I would base my final year project around this area, this is where I found the motivation to build a web-based platform designed for users to refine their coding skills through coding challenges and immediate feedback. The project was chosen with two goals: to enhance my programming language abilities and to create a resource that assists others in similar pursuits to mine.

Project Genesis

The inception of CodeHub was driven by a personal motivation to improve my software development skills. Throughout my 4 years of academic and personal experiences, I noticed a recurring challenge among peers and self-learners—the need for a practical, accessible means to practice coding and receive immediate feedback. This gap in educational tools sparked my interest in developing a platform that not only supports continuous learning but also makes the learning process engaging and user-centric.

Objectives

The primary objective of CodeChallenges is to provide a user-friendly environment where individuals can enhance their programming skills through a series of coding tests and challenges. The platform facilitates this by offering:

- Skill Enhancement: Allowing users to engage with various coding problems that grow their problem-solving skills and coding proficiency.

- Immediate Feedback: Implementing a system that evaluates code submissions against sample outputs to provide instant feedback and scores, aiding in immediate learning and improvement.

- Preparation for Professional Challenges: Equipping users with the necessary experience to excel at technical interviews and coding assessments commonly required in the tech industry.

General Issues

Along with the development of this project, numerous challenges came with it that tested both my technical and problem-solving abilities. This section outlines the significant issues I came across during the project's development, and how these issues were addressed. By sharing the challenges and solutions implemented, I hope to show the complexity involved in carrying out a user-friendly coding platform and the practical approaches taken to overcome these challenges.

Problems encountered, and how they were resolved

Deciding on a Color Scheme and Brand Identity

When developing a brand new platform, a common issue is how you want the website to look visually. Finding the right colour scheme and the brand logo is crucial in assuring the platform is not only attractive but also aligns with the intended audience and purpose

- Problem: Choosing a colour scheme and overall brand identity that aligns with the platform's objective
- Solution: Conducted research on websites similar to mine, learning the colours they use together, the design, the fonts and so on helped me get started with how I wanted my website to look.

Logo and Interface Design

The logo serves as the first point of engagement and is important for the brand's identity, choosing a logo that represented and matched my website took a lot of time and careful choosing.

- Problem: Creating a logo and user interface that is not only functional but also engaging and representative of the website's purpose
- Solution: Further research and a trial and error technique until I found one I believed represented my website well.

Platform Architecture and Technology Stack Selection

Choosing the right architecture and technology stack is a decision that has to be made carefully. Before developing my website, I had to think about what I wanted this website to accomplish, and what technologies I would need before starting to build the website

- Problem: Deciding on the appropriate technologies and frameworks to use for backend and frontend development that would support the intended functionality and scalability of the website
- Solution: Tested and compared different technologies, considering factors like useability, ease of integration and scalability.

Development Environment and IDE Setup

Carefully choosing and setting up an effective development environment tailored to my website's functionality was essential for productivity and efficiency. The IDE I chose was Visual Studio Code, because of not only my familiarity with it, but its User interface/user experience, its ability to install packages from the built-in terminal, and its overall efficiency for coding.

- Problem: Finding and setting up an IDE that supports all the necessary languages and tools needed for the project
- Solution: Trying out different IDEs for features like efficiency, UX, ease of use, and debugging tools, eventually finding an environment that meets nearly all development needs.

Building an Integrated Development Environment (IDE) into CodeHub

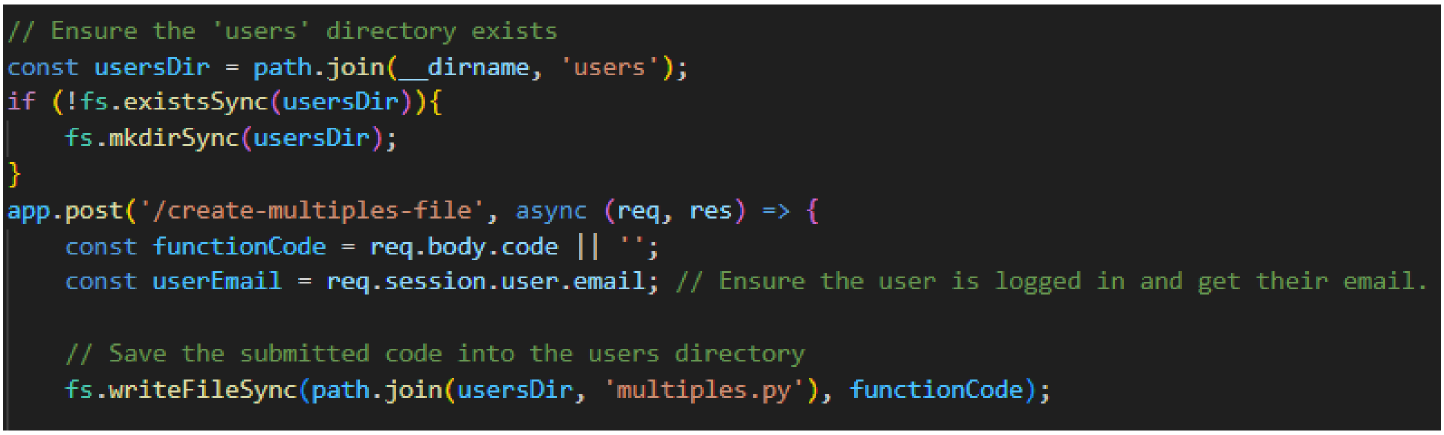
Integrating an IDE into my website proved to be my first coding problem, and was also the cornerstone feature for the rest of my website. Trying to find support online for building an IDE that users could write, test and submit code directly within the browser proved to be a challenge and time-consuming.

- Problem: Early attempts to build a custom IDE from scratch caused significant difficulties due to the complexity of developing features like code compiling, syntax highlighting, and error handling.
- Solution: After several attempts to evaluate third-party options. I found CodeMirror, an IDE library known for its broad language support and customization options. This library gave me the foundation I needed to start creating my IDE. CodeMirror was embedded into my website, and linked with the backend systems to manage code execution and user submissions

Saving User Submissions as .py Files

Automating the assessment of user-submitted code required a system of dynamically converting text input into executable Python files. This function would prove to be fundamental in the further development of the website, as it served as the foundation and beginning for the automated testing and validation of user solutions.

- Problem: A crucial step in automating the code was to save user-submitted into '.py' files, which required a reliable system to handle file creation and management
- Solution: I created a backend system that captures user code from the front end, and automatically generates the '.py' files and stores them in a folder for testing.



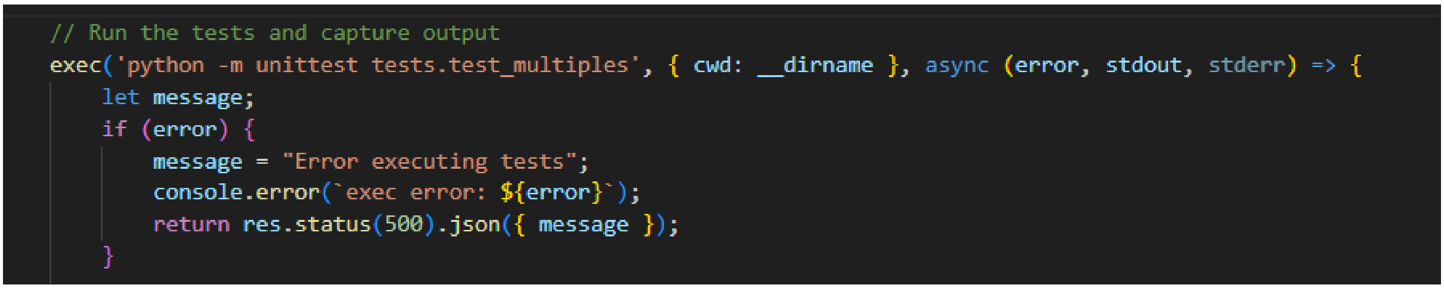
```
// Ensure the 'users' directory exists
const usersDir = path.join(__dirname, 'users');
if (!fs.existsSync(usersDir)){
    fs.mkdirSync(usersDir);
}
app.post('/create-multiples-file', async (req, res) => {
    const functionCode = req.body.code || '';
    const userEmail = req.session.user.email; // Ensure the user is logged in and get their email.

    // Save the submitted code into the users directory
    fs.writeFileSync(path.join(usersDir, 'multiples.py'), functionCode);
```

Figure 0.1: Save user file as .py

Validating User Submissions with Test Suites

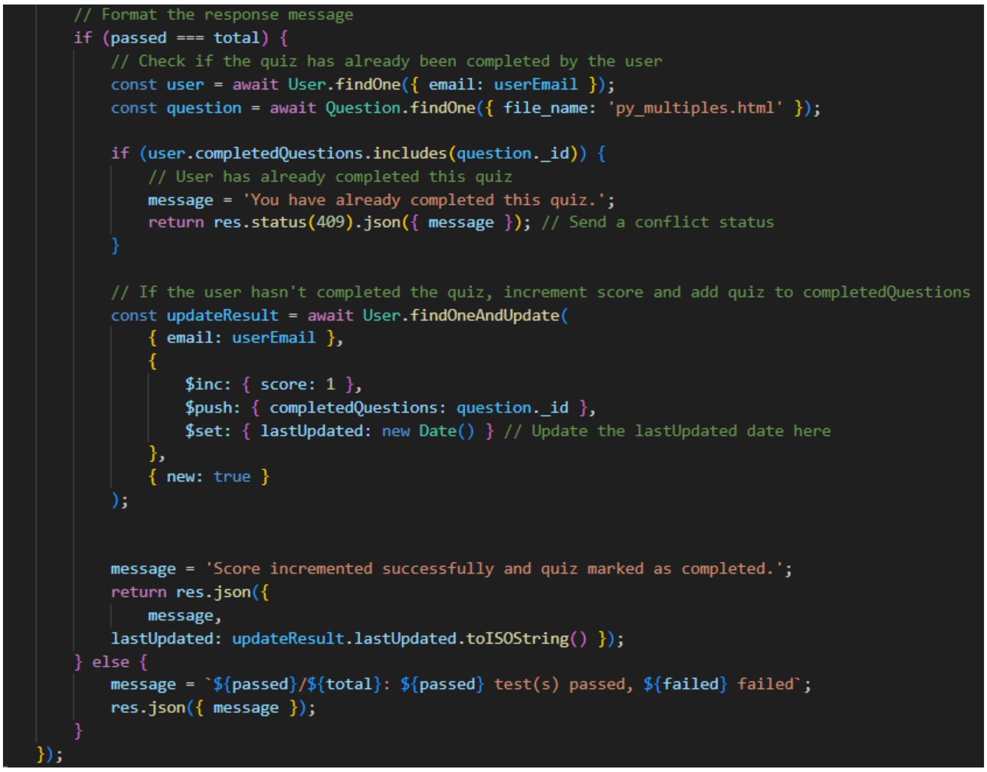
Ensuring the correctness of user submissions was paramount in maintaining the educational aspect of the website. Test suites were implemented to prevent users from cheating or ‘hard-coding’ answers into the compiler.

- Problem: Trying to ensure the quality of user submissions by testing the code against predefined test suites linked to the question being attempted.
- Solution: I created a series of test suites specific to each coding challenge. For example, if a user was taking the challenge of the multiple, their code would be saved as a ‘multiples.py’ file, and it tested with the test suites in the ‘test_multiples.py’ file. The system then evaluates whether the user’s solution meets the criteria before incrementing the user’s score only when all tests have passed. This automated system rewards users for their problem-solving skills, promoting a learning environment that emphasizes understanding the code over straight solutions.



```
    // Run the tests and capture output
    exec('python -m unittest tests.test_multiples', { cwd: __dirname }, async (error, stdout, stderr) => {
        let message;
        if (error) {
            message = "Error executing tests";
            console.error(`exec error: ${error}`);
            return res.status(500).json({ message });
        }
```

Figure 0.2: Execute unit test



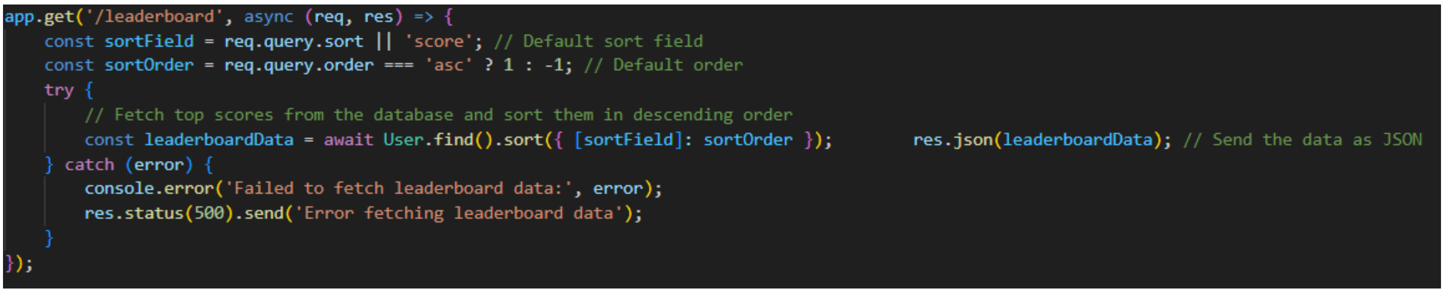
```
        // Format the response message
        if (passed === total) {
            // Check if the quiz has already been completed by the user
            const user = await User.findOne({ email: userEmail });
            const question = await Question.findOne({ file_name: 'py_multiples.html' });

            if (user.completedQuestions.includes(question._id)) {
                // User has already completed this quiz
                message = 'You have already completed this quiz.';
                return res.status(409).json({ message }); // Send a conflict status
            }

            // If the user hasn't completed the quiz, increment score and add quiz to completedQuestions
            const updateResult = await User.findOneAndUpdate(
                { email: userEmail },
                {
                    $inc: { score: 1 },
                    $push: { completedQuestions: question._id },
                    $set: { lastUpdated: new Date() } // Update the lastUpdated date here
                },
                { new: true }
            );


            message = 'Score incremented successfully and quiz marked as completed.';
            return res.json({
                message,
            lastUpdated: updateResult.lastUpdated.toISOString() });
        } else {
            message = `${passed}/${total}: ${passed} test(s) passed, ${failed} failed`;
            res.json({ message });
        }
    });
```

Figure 0.3: Incrementing Score

Implementing a Dynamic Leaderboard

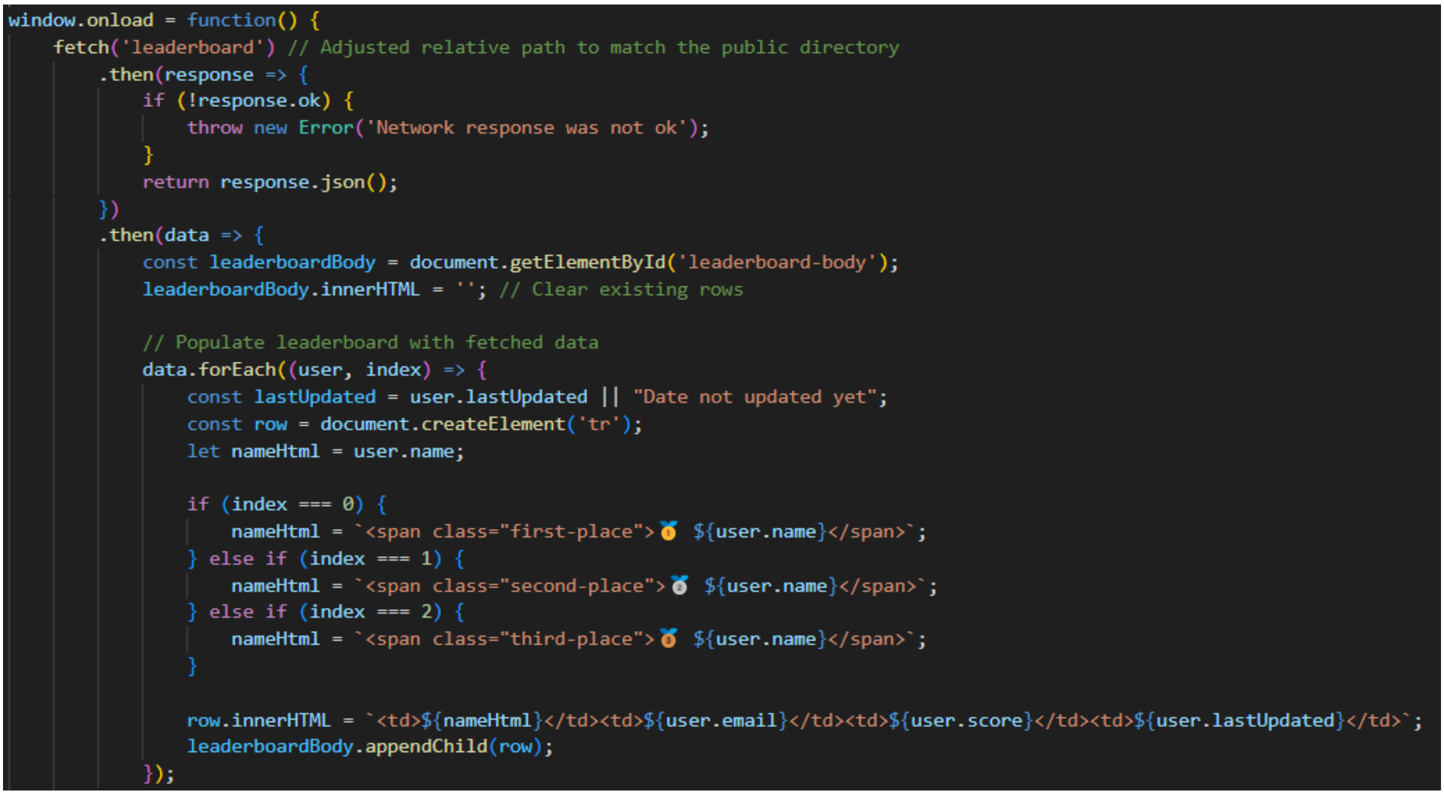
Implementing an interactive leaderboard presented some challenges, primarily related to how to store the user's progress and real-time updating. The leaderboard was designed to promote a competitive and engaging environment by showcasing other user achievements and progress. This feature required integration with the backend database MongoDB to fetch and display data accurately and promptly. The goal was to ensure that the leaderboard was not only informative but also encouraged users to engage more with the challenges by seeing their rankings in comparison to others.

- Problem: Creating a live leaderboard involving linking user data (name, email, score, and date challenge finished) dynamically to a 'leaderboard.html' page, which needed to reflect real-time updates as scores changed.
- Solution: I used a combination of server-side scripting and AJAX to fetch and update the leaderboard data without requiring page reloads. The system pulls updated data from the database, and displays it on the leaderboard page, ensuring the leaderboard reflects the scores in order of highest to lowest.



```
app.get('/leaderboard', async (req, res) => {
    const sortField = req.query.sort || 'score'; // Default sort field
    const sortOrder = req.query.order === 'asc' ? 1 : -1; // Default order
    try {
        // Fetch top scores from the database and sort them in descending order
        const leaderboardData = await User.find().sort({ [sortField]: sortOrder });        res.json(leaderboardData); // Send the data as JSON
    } catch (error) {
        console.error('Failed to fetch leaderboard data:', error);
        res.status(500).send('Error fetching leaderboard data');
    }
});
```

Figure 0.4: app.get /leaderboard



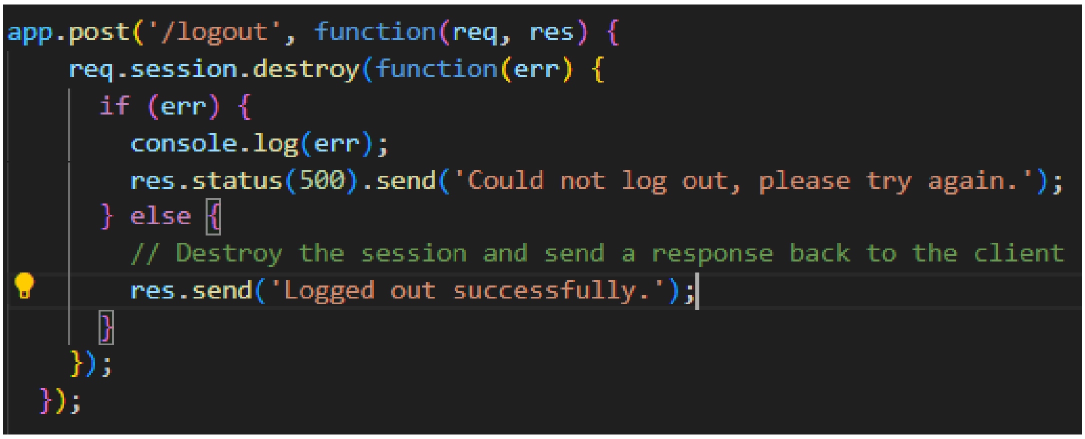
```
window.onload = function() {
    fetch('leaderboard') // Adjusted relative path to match the public directory
        .then(response => {
            if (!response.ok) {
                throw new Error('Network response was not ok');
            }
            return response.json();
        })
        .then(data => {
            const leaderboardBody = document.getElementById('leaderboard-body');
            leaderboardBody.innerHTML = ''; // Clear existing rows

            // Populate leaderboard with fetched data
            data.forEach((user, index) => {
                const lastUpdated = user.lastUpdated || "Date not updated yet";
                const row = document.createElement('tr');
                let nameHtml = user.name;

                if (index === 0) {
                    nameHtml = `<span class="first-place">🥇 ${user.name}</span>`;
                } else if (index === 1) {
                    nameHtml = `<span class="second-place">🥈 ${user.name}</span>`;
                } else if (index === 2) {
                    nameHtml = `<span class="third-place">🥉 ${user.name}</span>`;
                }

                row.innerHTML = `<td>${nameHtml}</td><td>${user.email}</td><td>${user.score}</td><td>${user.lastUpdated}</td>`;
                leaderboardBody.appendChild(row);
            });
```

Figure 0.5: Creating Leaderboard

Managing User Sessions and Secure Logout

Ensuring the security and integrity of user sessions was an important step to fix. To address this I had to create a logout mechanism to safeguard user privacy and security.

- Problem: Ensuring that user sessions were properly terminated when users logged out, this involved ensuring that all session information was completely cleared to prevent unauthorized access or lingering data after a user logged out.
- Solution: I implemented a logout feature that destroyed the user's sessions on the server and cleared all related data from the client's browser. This method ensured that once a user logged out, their session could not be reused.



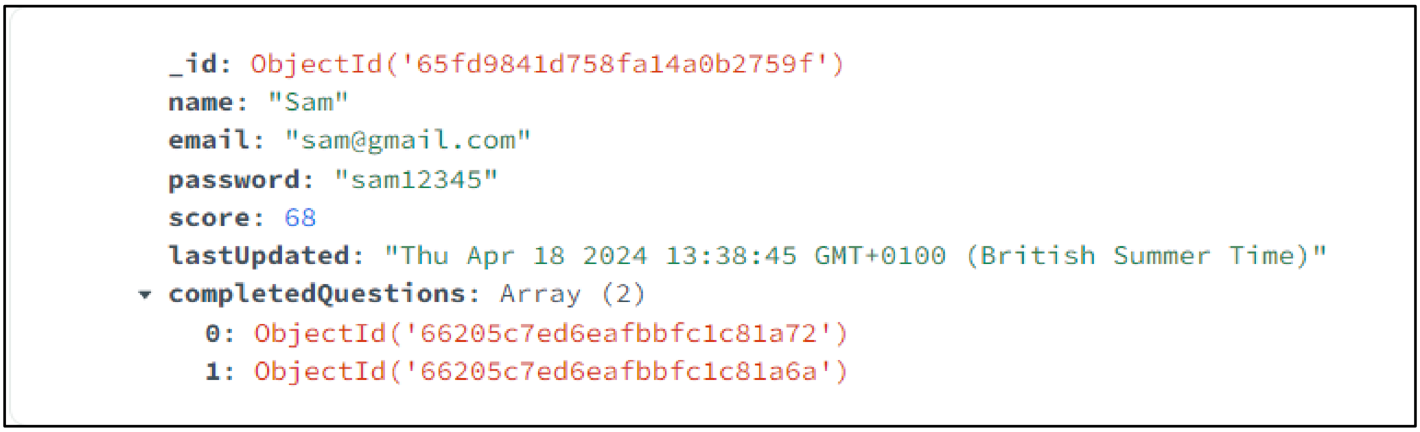
```
app.post('/logout', function(req, res) {
    req.session.destroy(function(err) {
      if (err) {
        console.log(err);
        res.status(500).send('Could not log out, please try again.');
      } else {
        // Destroy the session and send a response back to the client
        res.send('Logged out successfully.');
      }
    });
  });
```

Figure 0.6: Destroy user session

Preventing Repeated Score Increments for the Same Challenge

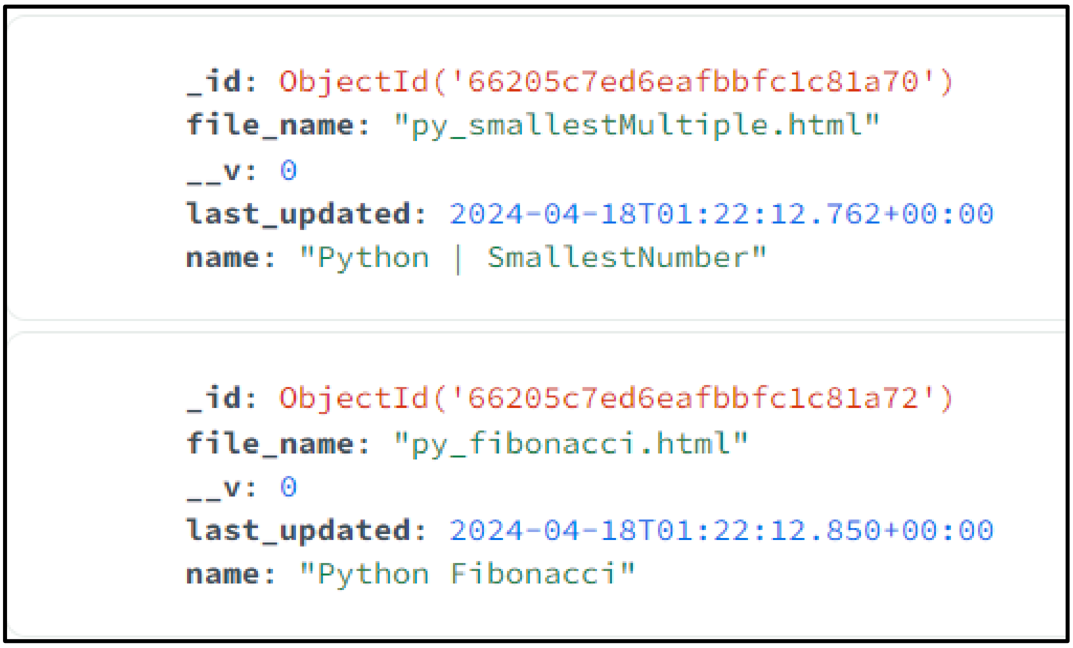
To maintain a fair and competitive environment, I needed to create a feature that would prevent users from gaining additional points by repeatedly completing the same coding challenges, The solution involved creating a system that

- Problems: Trying to promote a fair learning environment and prevent score manipulation to stop users from earning multiple scores from the same challenge.
- Solution: Created a system that tracks the user that is currently logged in, when a user completes a question for the first time, the ID of that quiz is then stored within the user's table located with the user that passed the test. If the user tries to repeat a question they currently have the ID of stored in the database, they are told they have already gained a score from this question



```
_id: ObjectId('65fd9841d758fa14a0b2759f')
name: "Sam"
email: "sam@gmail.com"
password: "sam12345"
score: 68
lastUpdated: "Thu Apr 18 2024 13:38:45 GMT+0100 (British Summer Time)"
▾ completedQuestions: Array (2)
    0: ObjectId('66205c7ed6eafbbfc1c81a72')
    1: ObjectId('66205c7ed6eafbbfc1c81a6a')
```

Figure 0.7: User Database Record



```
_id: ObjectId('66205c7ed6eafbbfc1c81a70')
file_name: "py_smallestMultiple.html"
__v: 0
last_updated: 2024-04-18T01:22:12.762+00:00
name: "Python | SmallestNumber"

_id: ObjectId('66205c7ed6eafbbfc1c81a72')
file_name: "py_fibonacci.html"
__v: 0
last_updated: 2024-04-18T01:22:12.850+00:00
name: "Python Fibonacci"
```

Figure 0.8: Question Database Record

What I achieved

Comprehensive User Interface and Accessibility Enhancements

By carefully selecting the colour scheme, creating a responsive layout, and ensuring the platform is accessible on various devices. The design considerations taken ensure that the platform is not only visually appealing but also user-friendly and easy to navigate around, making the process for users more efficient and enjoyable

Integrated Development Environment (IDE) Integration

By incorporating CodeMirror into CodeHub, I provided users with a user-friendly environment for coding directly on the website. This IDE supports syntax highlighting, error handling, and multiple programming languages. The customization of CodeMirror allowed me to integrate it with the overall design of the CodeHub, ensuring users have a consistent and intuitive coding experience

Dynamic Code Testing and Evaluation

Integrated within the development environment, I implemented a system that captures user input, converts it into a '.py' file, and tests it against predefined test suites related to the file. This setup ensures that if all tests are passed, users' scores are automatically incremented and updated in the database and the leaderboard page. This provides immediate feedback on their coding proficiency and encourages learning through trial and error

Real-Time Leaderboard Implementation

I created a dynamic leaderboard that not only displays users' names, emails, and scores but also updates in real time as a user receives a new score. This leaderboard pulls data directly from the MongoDB database, ensuring all information is current and accurate to reflect the competitive landscape of the platform.

Challenge Completion Tracking

To promote the learning experience for users, I implemented a feature that prevents users from gaining additional scores from challenges they have already completed. This feature not only maintains the integrity of the scoring system but encourages users to explore new challenges and gain a broad experience across various aspects of programming

Full-Functioning Website Development

Overall, I created a fully functional website dedicated to taking Python coding challenges. This platform serves as a comprehensive resource for users to engage in practical coding exercises tailored to enhance their programming skills systematically

What I Did Not Achieve

While developing CodeHub, there were many envisioned features I could not implement within the project timeline. Some significant reasons for these shortcomings were:

Time Constraints: Trying to balance working on this project alongside other academic projects and deadlines proved challenging, balancing these commitments significantly limited the amount of time available to dedicate to expanding CodeHub's features

Issue Resolution: A considerable portion of the project timeline was consumed by identifying and resolving unexpected technical issues that arose during the development phases.

Resource Limitations: Since the project was done entirely by myself only, the range of tasks that could be accomplished at once was naturally restricting.

Despite the shortcomings, some features I would like to add after the project deadline in my own time are:

Broader Question Repository

I'd like to add more available challenges for users to take, increasing the amount of possible questions helps keep users returning to the website and improving their skills

Support for Additional Programming Languages

I'd like to add more languages to the website for users to try, I had the idea of adding Java and C++ alongside the process with Python, but I took it upon myself to focus on one language to get completely functional rather than having multiple languages only partially working.

Enhanced Leaderboard Features

I'd intended to add more features to the leaderboard page, features like recording the time it takes users to finish a challenge would add a competitive aspect to the website, motivating users to not only correct their code but optimize their problem-solving speed

User Progress Tracking Page

I planned to develop a page where the logged-in user could view all the challenges they have completed, offering them a clear view of their progress. This feature would have enhanced user engagement by providing a sense of achievement and encouraging continued participation

Detailed Feedback on Test Suites

Improving the feedback provided by the test suites was another objective. Detailed feedback such as specifying the number of tests passed (e.g., "4/5 tests passed") was the idea. This would help users understand whether the code they are writing was going in the right direction or not.

Password Security

I intend to add more layers of security to the user's passwords in the database. For the project, I have not added this feature since I am not yet ready to deploy the website publicly; but will add security measures like password encryption using salt and hash round on the passwords.

What I learned

The development journey of CodeHub was not only about building a software platform but also about personal and professional growth as a software developer which provided me with many valuable insights and skills. Some key learning experiences I took from this experience are:

Project Management

Managing a complex software project from start to completion gave me valuable insights into project planning, scope management, and task prioritization. I learned how to break down large tasks into smallest, more manageable tasks, fix them, and then implement these small changes back into my main project.

Time Management

Balancing the development of CodeHub with academic responsibilities and other projects was challenging. This experience of working on a project where we are given a lengthy time to do and working with assignments with shorter deadlines taught me when and how to balance working between these projects.

Importance of taking breaks

One valuable lesson I learned during the development of CodeHub is the importance of taking short breaks. There were times when I found myself working late into the night, sometimes until 3 am, trying to solve a problem for hours on end. Surprisingly, after taking a day break and returning with a fresh mind, I was often able to solve the same problem in a short time. This experience taught me that continuous work without breaks can lead to a decline in productivity and tunnel vision. Regular breaks, especially when stuck on a problem for a long time, can significantly enhance problem-solving skills. This practice of pacing myself and managing mental resources wisely is something that stood out to me drastically.

Problem-Solving and Debugging

Throughout the project, I encountered and resolved multiple technical challenges. This process helped improve my problem-solving skills and taught me effective debugging techniques, especially in a full-stack environment where issues can range across different layers of the application. With all this, I also learned to be a more confident software developer, gaining the confidence to know I can solve problems that arise.

Technical Proficiency

Through the challenges I encountered during the development process, I deepened my understanding of web development principles, backend architecture, and database management. Overcoming issues such as optimizing code execution and integrating external libraries improved my problem-solving skills and expanded my technical toolkit

Automated Testing and Continuous Integration

Developing automated tests for coding challenges helps me understand the role of test-driven development (TDD) and continuous integration (CI) in maintaining code functionality throughout the project lifecycle

Effective Communication

Communicating with my supervisor each week, and seeking another person's input into the project proved very valuable for gathering new ideas, gathering requirements for future features, and resolving issues effectively

Full Stack Development Skills

Throughout this project, I gained comprehensive experience with full-stack development, managing both the front-end and back-end aspects of the platform. This included everything from designing the user interface to handling server-side logic and database management. The integration of technologies such as MongoDB, Express.js, and Node.js, along with the front-end frameworks, improved my understanding of how various levels of development work together in a real-world application

What I would do differently if starting again

Now that I am at the end of the project deadline, reflecting on the development of CodeHub there are several aspects I would like to approach differently if starting the project again. These changes would help to improve different aspects of the project like the platform's functionality, time planning, and overall project management.

Improved Initial Planning

If I were to start again, I would invest more time into the initial planning and design phase to better understand what route I would go with for building the website, using tools like wireframes, user flow diagrams, and a better to-do list ta han I had that aligns with the project goals. Better planning would save time throughout the whole lifecycle of the project.

Expand on the Language Support

I would like to focus more on making the website multi-language early in the development process. This feature would cater to a broader range of users from the beginning and avoid the complexity of adding new languages later on

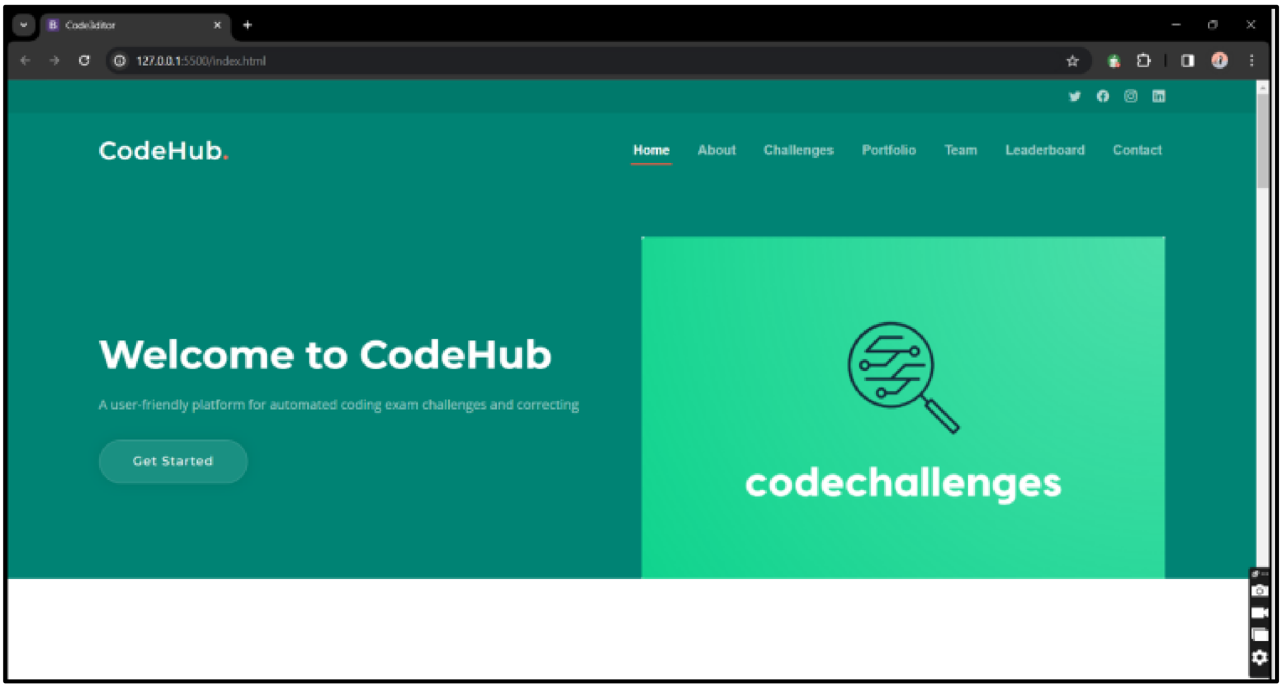
Strategic Breaks and Time Management

Having learned and noticed the importance of taking breaks has had on the project, I would set up structured breaks and work to a more disciplined work schedule. This approach would help maintain high productivity without leading to burnout.

Technical Issues

Throughout the development of CodeHub, the project underwent many evolutionary designs from its initial design to the final product. In this section I will address the differences between the earlier functionalities and the actual system deployed, highlighting the adjustments made to address these issues. The following points detail these shifts in design and implementation and provide an insight into the dynamic nature of software development.

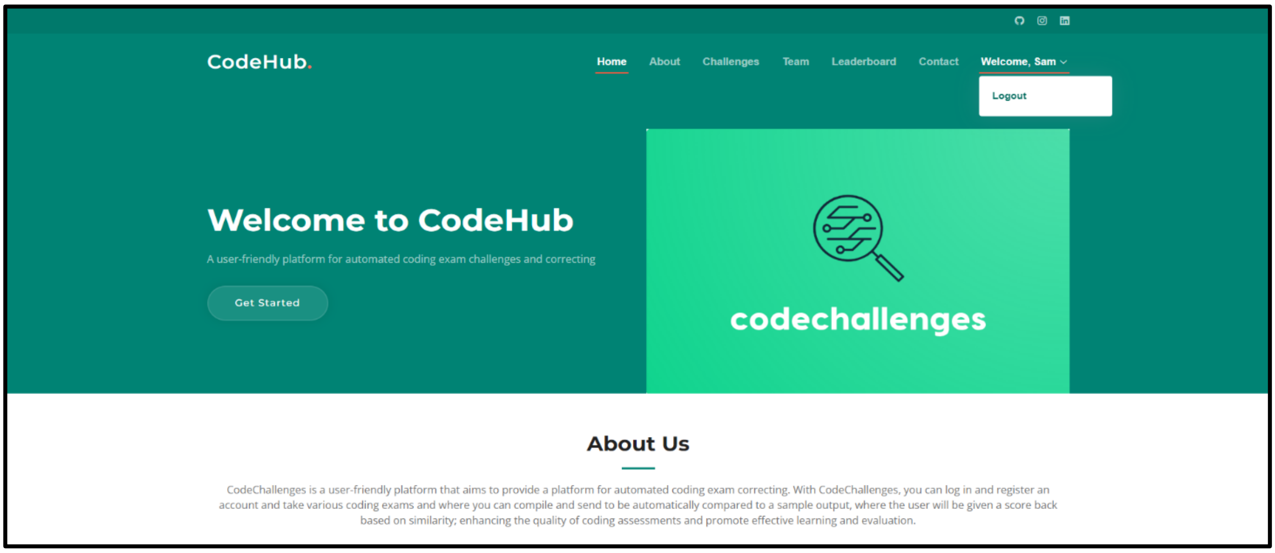
Report of differences from early design



Welcome to CodeHub

A user-friendly platform for automated coding exam challenges and correcting

Figure 0.9: Early homepage development



Welcome to CodeHub

A user-friendly platform for automated coding exam challenges and correcting

About Us

CodeChallenges is a user-friendly platform that aims to provide a platform for automated coding exam correcting. With CodeChallenges, you can log in and register an account and take various coding exams and where you can compile and send to be automatically compared to a sample output, where the user will be given a score back based on similarity; enhancing the quality of coding assessments and promote effective learning and evaluation.

Figure 1.0: Final homepage development

User Authenticated Update

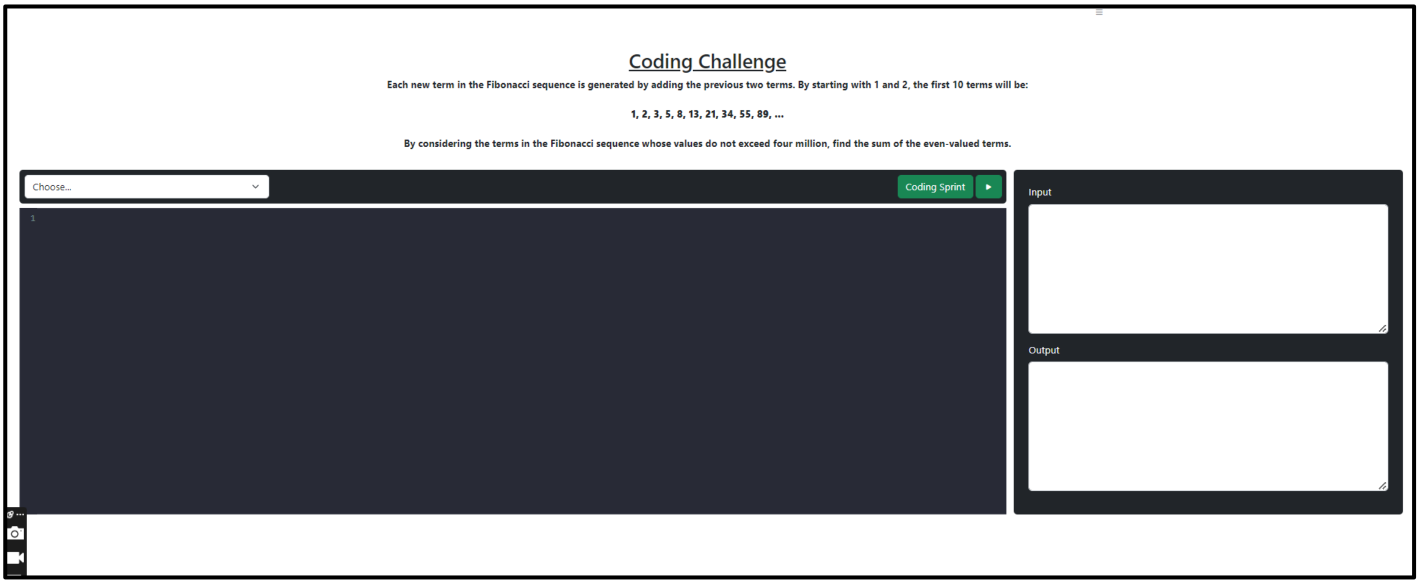
The new version shows a user login system with a personalized welcome message and a logout option, indicating the implementation of user session management, and allowing for a more tailored experience and the ability to save and track user progress and scores

Navigation Bar Improvements

The navigation bar receives a few minor changes, giving a more clear view of the sections of the website. The addition of dropdown menus user the user's name suggests enhanced navigational functionality, enabling the users to easily access their profiles and other personalized areas of the site

Responsive Design

Although both versions are responsive, the newer version shows a better-defined layout with elements spaced better, which indicates improvements in responsive design practices.



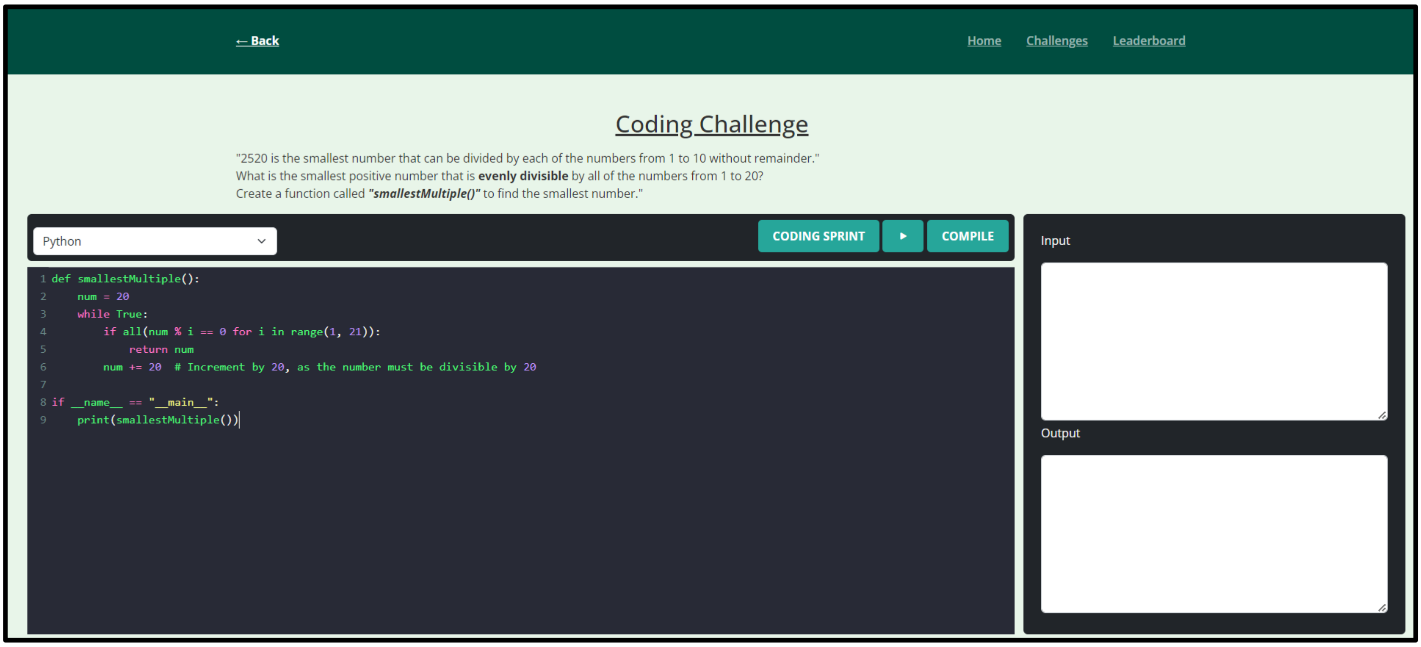
Coding Challenge

Each new term in the Fibonacci sequence is generated by adding the previous two terms. By starting with 1 and 2, the first 10 terms will be:

1, 2, 3, 5, 8, 13, 21, 34, 55, 89, ...

By considering the terms in the Fibonacci sequence whose values do not exceed four million, find the sum of the even-valued terms.

Figure 1.1: Early compiler page development



Coding Challenge

"2520 is the smallest number that can be divided by each of the numbers from 1 to 10 without remainder."
What is the smallest positive number that is **evenly divisible** by all of the numbers from 1 to 20?
Create a function called ***"smallestMultiple()"*** to find the smallest number."

```python
def smallestMultiple():
    num = 20
    while True:
        if all(num % i == 0 for i in range(1, 21)):
            return num
        num += 20  # Increment by 20, as the number must be divisible by 20

if __name__ == "__main__":
    print(smallestMultiple())
```

Figure 1.2: Final compiler page development

Visual Design and Color Scheme

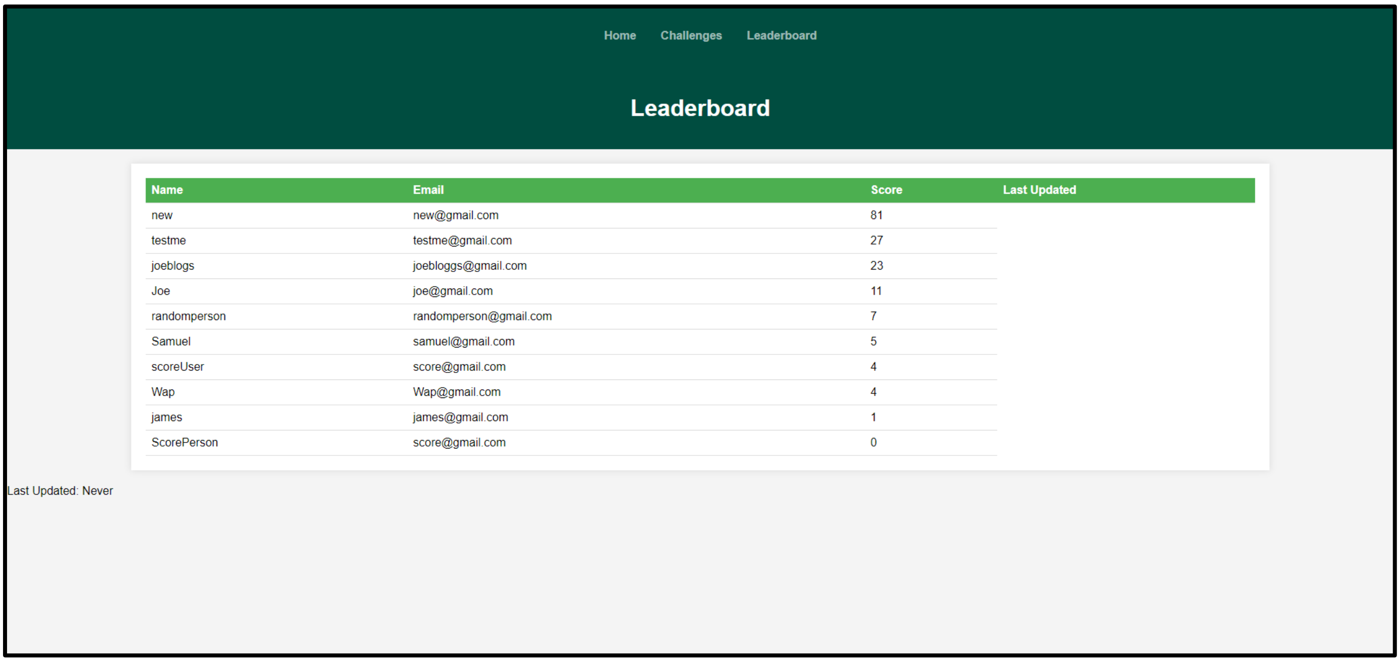
The newer version I created with the green background has a more vibrant and engaging colour scheme, I chose this colour due to the alignment with the CodeHub branding. The idea is to use consistent brand colours across the platform to enhance user recognition and brand recall.

Layout and Readability

The layout of the challenge description and the coding area in the new version has drastically
Improved the readability with a cleaner, more presentable compiler page. The question prompt is also clearer to the user. Overall, there is a clear distinction between the problem statement, code editor and, the input/output sections

User Interface (UI) Elements

I enhanced different UI components, such as buttons labelled "Compile" to provide a clear path to where the user will go to test their code.

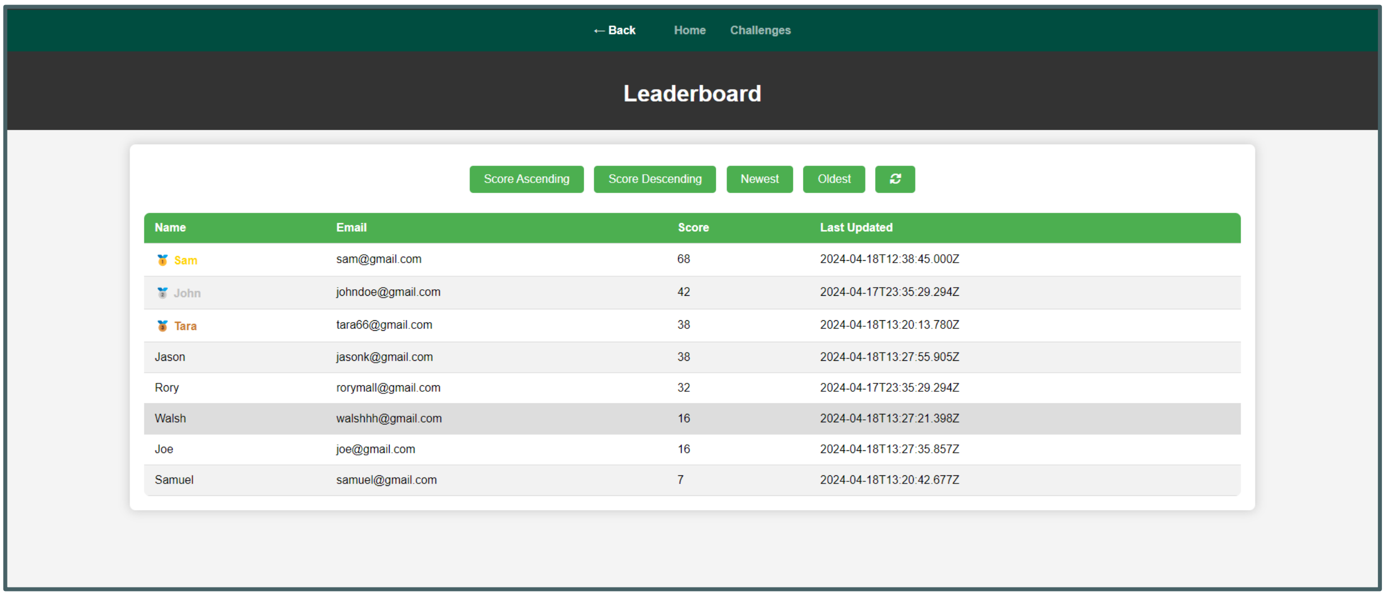


Leaderboard

Name	Email	Score	Last Updated
new	new@gmail.com	81	
testme	testme@gmail.com	27	
joeblogs	joebloggs@gmail.com	23	
Joe	joe@gmail.com	11	
randomperson	randomperson@gmail.com	7	
Samuel	samuel@gmail.com	5	
scoreUser	score@gmail.com	4	
Wap	Wap@gmail.com	4	
james	james@gmail.com	1	
ScorePerson	score@gmail.com	0	

Last Updated: Never

Figure 1.3: Early leaderboard development



Leaderboard

Name	Email	Score	Last Updated
🥇 Sam	sam@gmail.com	68	2024-04-18T12:38:45.000Z
🥈 John	johndoe@gmail.com	42	2024-04-17T23:35:29.294Z
🥉 Tara	tara66@gmail.com	38	2024-04-18T13:20:13.780Z
Jason	jasonk@gmail.com	38	2024-04-18T13:27:55.905Z
Rory	rorymall@gmail.com	32	2024-04-17T23:35:29.294Z
Walsh	walshhh@gmail.com	16	2024-04-18T13:27:21.398Z
Joe	joe@gmail.com	16	2024-04-18T13:27:35.857Z
Samuel	samuel@gmail.com	7	2024-04-18T13:20:42.677Z

Figure 1.4: Final leaderboard development

Inclusion of Last Updated Timestamp

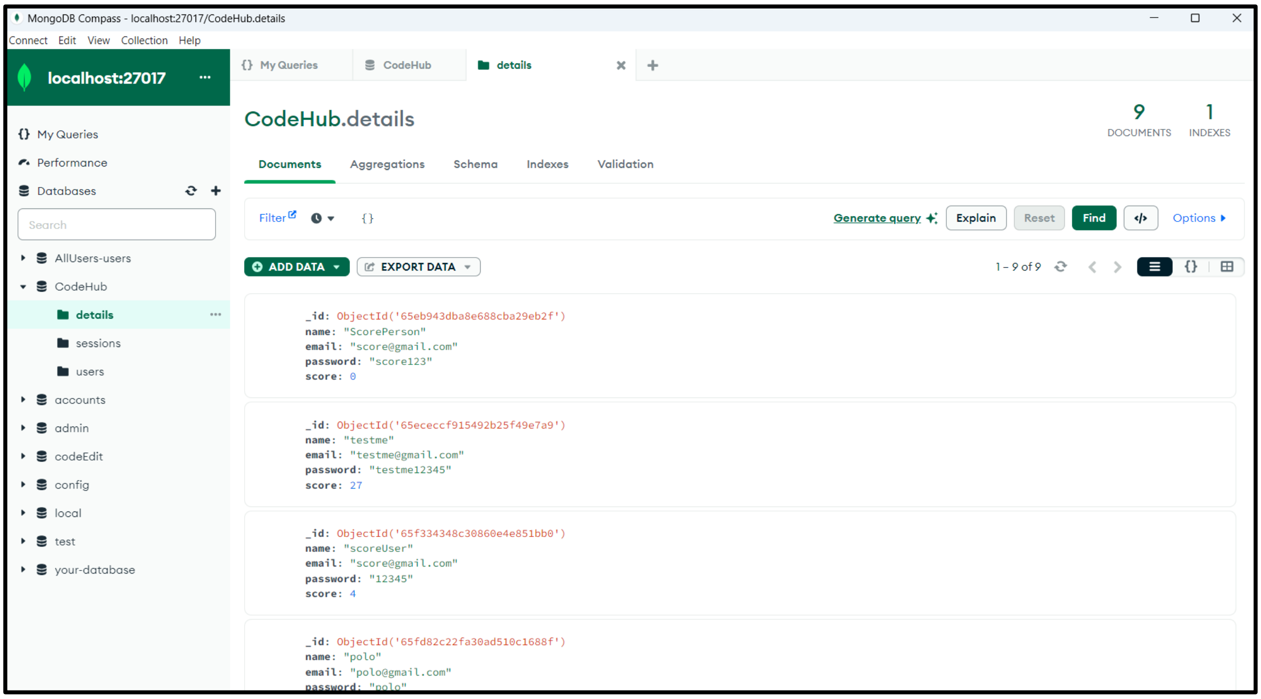
1) **Old Version:** Basic, and did not display user's updated date when necessary
2) **New Version:** Includes a 'Last Updated' column, providing transparency about when a score was last updated. This new addition gives users up-to-date information and reflects the dynamics associated with the leaderboard.

Visual Design and Layout

1) **Old Version:** Has a simple, basic design and background with a plain table structure
2) **New Version:** The new layout includes a clean, modern design with clear separation between columns to help users with readability. The use of colour was done to highlight the column headers and sorting options to improve the visual display and make it easier for users to track the page

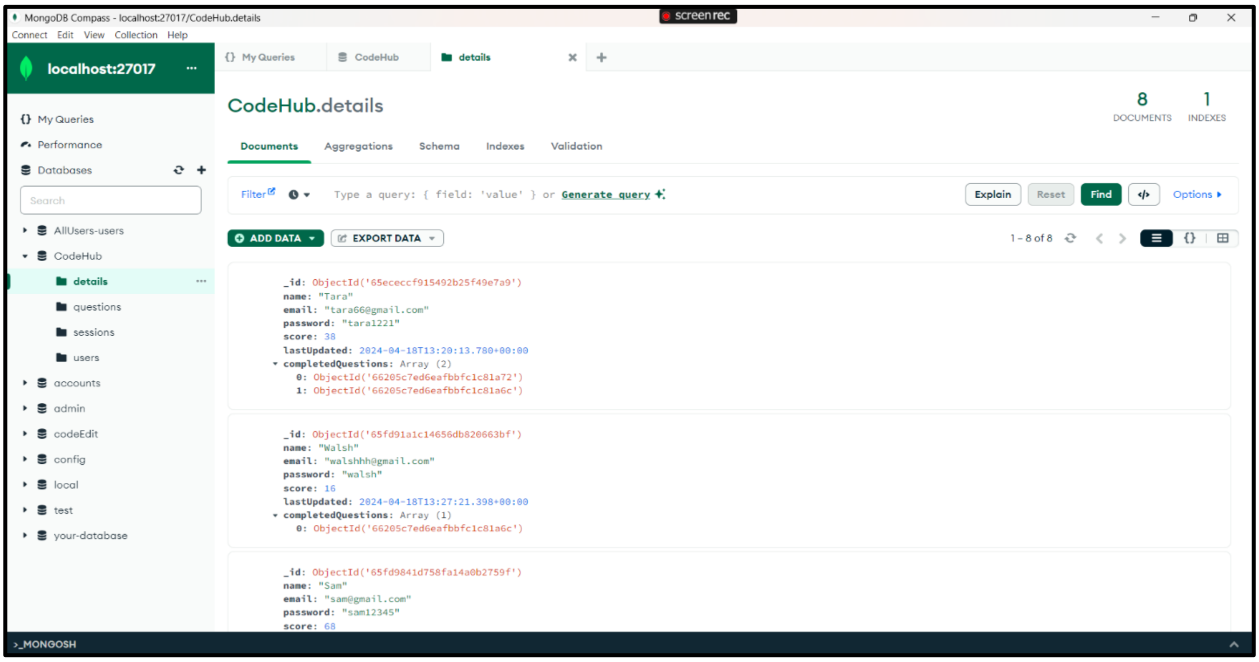
Sorting Functionality

1) **Old Version:** Leaderboard is static, and lacks the options to sort the data displayed.
2) **New Version:** Adding sorting buttons offers a new user interaction, allowing users to view who would be at the top of the table. This feature again promotes the competitiveness of viewing other users at the top and aims to get the user to engage with the website more.



```
_id: ObjectId('65eb943dba8e688cba29eb2f')
name: "ScorePerson"
email: "score@gmail.com"
password: "score123"
score: 0

_id: ObjectId('65ececcf915492b25f49e7a9')
name: "testme"
email: "testme@gmail.com"
password: "testme12345"
score: 27

_id: ObjectId('65f334348c30860e4e851bb0')
name: "scoreUser"
email: "score@gmail.com"
password: "12345"
score: 4

_id: ObjectId('65fd82c22fa30ad510c1688f')
name: "polo"
email: "polo@gmail.com"
password: "polo"
```

Figure 1.5: Early stage of database records

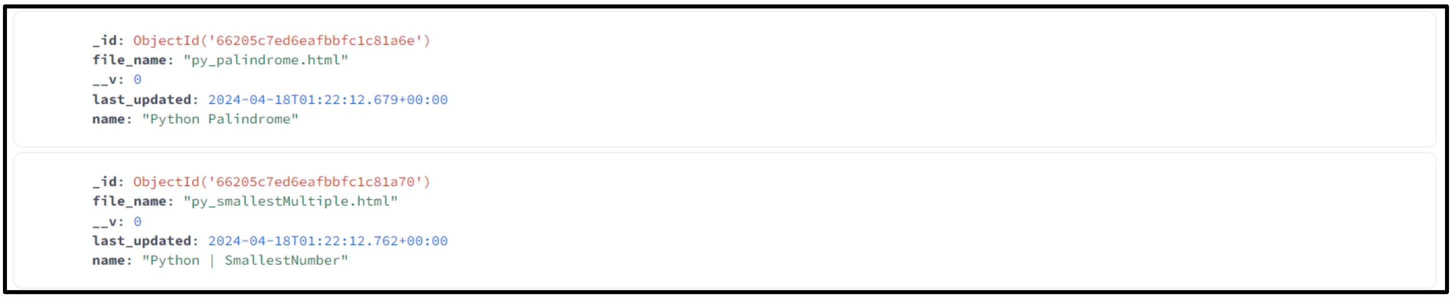


```
_id: ObjectId('65ececcf915492b25f49e7a9')
name: "Tara"
email: "tara66@gmail.com"
password: "tara1221"
score: 38
lastUpdated: 2024-04-18T13:20:13.780+00:00
completedQuestions: Array (2)
  0: ObjectId('66205c7ed6eafbbfc1c81a72')
  1: ObjectId('66205c7ed6eafbbfc1c81a6c')

_id: ObjectId('65fd91a1c14656db820663bf')
name: "Walsh"
email: "walshhh@gmail.com"
password: "walsh"
score: 16
lastUpdated: 2024-04-18T13:27:21.398+00:00
completedQuestions: Array (1)
  0: ObjectId('66205c7ed6eafbbfc1c81a6c')

_id: ObjectId('65fd9841d758fa14a0b2759f')
name: "Sam"
email: "sam@gmail.com"
password: "sam12345"
score: 68
```

Figure 1.6: Final stage of database records

User Data Structure

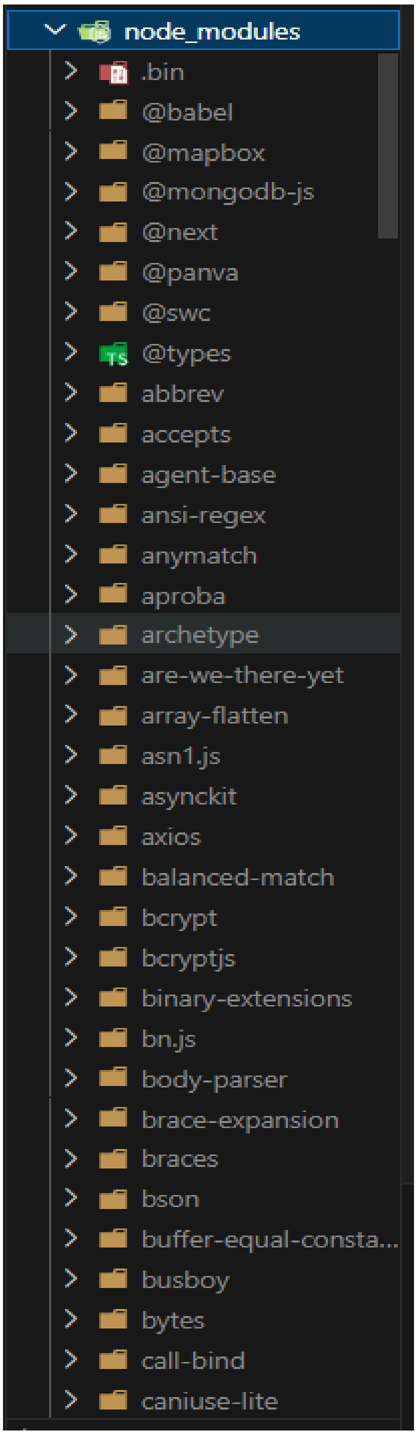
1) **Old Version:** The user document contains basic fields like name. email, password, and score
2) **New Version:** Additional detail has been added with fields like 'lastUpdated' which records the date and time a user last updated their score value. The 'completedQuestions' field retrieves the ID of the question the user has completed, this is done to prevent users from attempting the same question over and over again.



```
_id: ObjectId('66205c7ed6eafbbfc1c81a6e')
file_name: "py_palindrome.html"
__v: 0
last_updated: 2024-04-18T01:22:12.679+00:00
name: "Python Palindrome"

_id: ObjectId('66205c7ed6eafbbfc1c81a70')
file_name: "py_smallestMultiple.html"
__v: 0
last_updated: 2024-04-18T01:22:12.762+00:00
name: "Python | SmallestNumber"
```

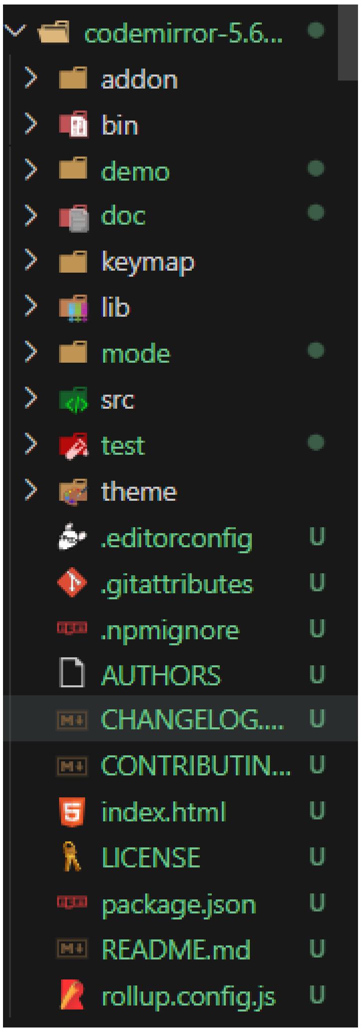
Figure 1.7: Questions database records

Module Descriptions



```
node_modules
  .bin
  @babel
  @mapbox
  @mongodb-js
  @next
  @panva
  @swc
  @types
  abbrev
  accepts
  agent-base
  ansi-regex
  anymatch
  aproba
  archetype
  are-we-there-yet
  array-flatten
  asn1.js
  asynckit
  axios
  balanced-match
  bcrypt
  bcryptjs
  binary-extensions
  bn.js
  body-parser
  brace-expansion
  braces
  bson
  buffer-equal-consta...
  busboy
  bytes
  call-bind
  caniuse-lite
```

Figure 1.8: node_module



```
codemirror-5.6...
  addon
  bin
  demo
  doc
  keymap
  lib
  mode
  src
  test
  theme
  .editorconfig      U
  .gitattributes     U
  .npmignore         U
  AUTHORS            U
  CHANGELOG....      U
  CONTRIBUTIN...     U
  index.html         U
  LICENSE            U
  package.json       U
  README.md          U
  rollup.config.js   U
```

Figure 1.9: codemirror package

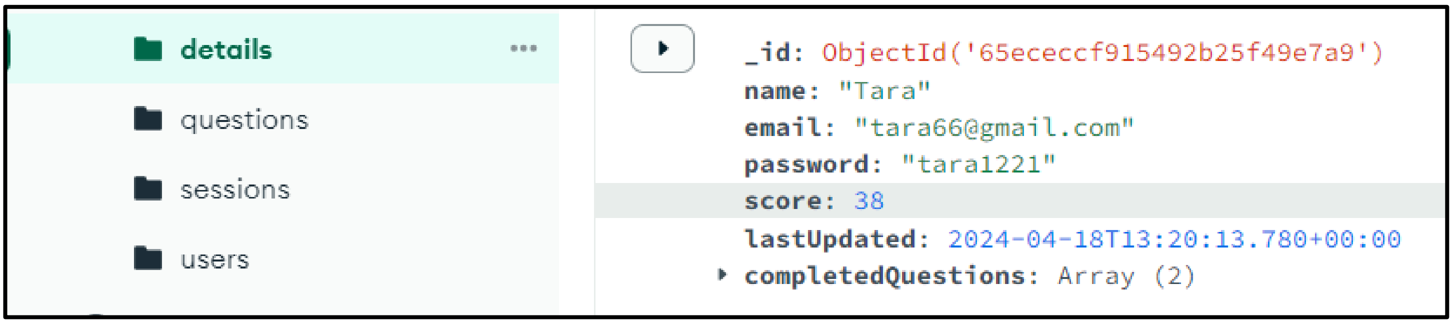
Codemirror

The “codemirror” folder is part of my project’s structure. It relates to the CodeMirror library, which is a versatile text editor implemented in JavaScript for the browser. It specializes in editing code and comes with several language models and add-ons that implement more advanced features and functions. I mainly prioritized CodeMirror for an in-browser coding experience, to allow users to write code from the webpage. This helped with files like ‘py_multiples.html’ where the CodeMirror library is used to give me a foundation to start building a functional code editor, with support for syntax highlighting.

Node_modules

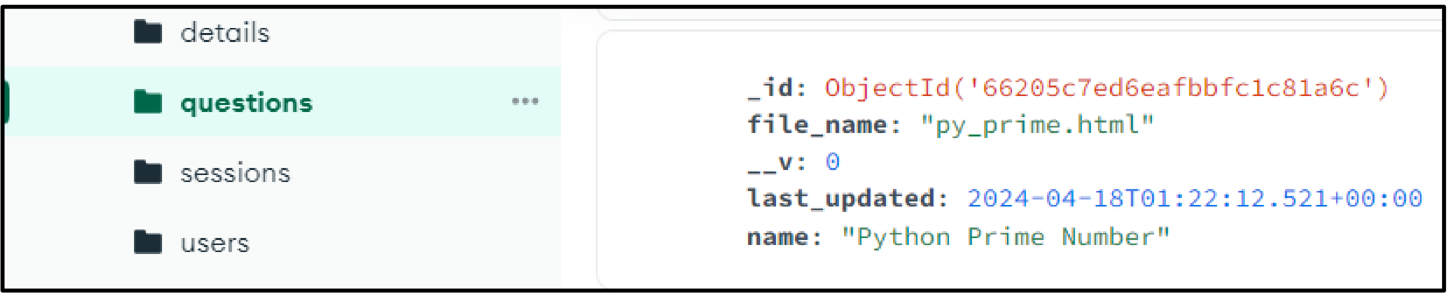
The “node_module” directory is generated when I use npm (Node Package Manager) to install dependencies for the Node.js application. Each subdirectory represents a package or module that my project depends on, from libraries, frameworks, and tools that help improve the functionality of my website.

Data Structures & Database Record Layout



```
_id: ObjectId('65ececcf915492b25f49e7a9')
name: "Tara"
email: "tara66@gmail.com"
password: "tara1221"
score: 38
lastUpdated: 2024-04-18T13:20:13.780+00:00
completedQuestions: Array (2)
```

Figure 2.0: User record field layout



```
_id: ObjectId('66205c7ed6eafbbfc1c81a6c')
file_name: "py_prime.html"
__v: 0
last_updated: 2024-04-18T01:22:12.521+00:00
name: "Python Prime Number"
```

Figure 2.1: Question record field layout

How data is structured and managed becomes foundational to the system’s overall performance. For my project, the infrastructure is illustrated through the use of MongoDB for database management, and several front-end technologies that interface with our server-side logic.

Data Structures

- **HTML Documents**
Files like ‘py_prime.html’, ‘leaderboard.html’, and ‘homepage.html’ are mainly structured with HTML tags, styled with CSS, and contain scripts for dynamic behaviours. The forms and interactive elements on these pages collect data from the user which is then sent to the server in a structured JSON format.

- **JSON Objects**
Interaction with the server involves JSON objects. In ‘signup.js’ when a user signs up, their username, email and password are sent as a JSON object in the body of a POST request. The server then responds with JSON-formatted messages that display the result.

Database Layout

The database is stored in MongoDB, a NoSQL database. The database uses two main tables, ‘details’ and ‘questions’

- **Collections**

‘questions’: This collection stores the coding challenges on the website. Each document has fields like ‘file_name’, ‘last_updated’, ‘name’, and ‘_id’

‘details’: This collection stores user details. The fields in this collection are ‘name’, ‘email’, ‘password, ’‘score’, ‘completedQuestions’, and ‘lastUpdated’. This structure allows for storing user credentials, users' performance, and progress within the website

Integration

The integration between the front end and the back end is evident through the use of API calls in the JavaScript files. These calls interact with the database by sending or requesting JSON data, which MongoDB can understand the process

Testing used to assess reliability

Ensuring the reliability of software, particularly for an educational tool where the integrity of the results directly impacts users’ learning outcomes. To ensure this, I carried out multiple testing protocols throughout the development lifecycle.

Test Suites for Code Validation

To prevent users from simply hardcoding or using other forms of cheating, writing comprehensive test suites that not only evaluate if the output is correct, but also adhere to the problem-solving constraints. These tests were designed to cover a broad range of inputs, including edge cases, to make sure that the code responds correctly.

Session Management and Security

From a security point of view, the application has a session management system feature. When a user logs out, a function is executed to ensure that the session is destroyed; preventing session highjacking or authorized access.

User Experience Feedback

Lastly, I asked a few peers to test my website for themselves, taking into consideration everything from the user interface, user experience, backend features, and so on. These tests helped me understand how the user navigates through the challenges, writes code, compiles their code, and receives the correct feedback

Conclusion

Throughout the development of the CodeHub website, I found that this Final Year Project served not only as a technical challenge but also as a learning and personal development experience.
CodeHub successfully integrates an intuitive user interface, a dynamic Integrated Development Environment (IDE), code testing and evaluation, and a real-time leaderboard, which collectively creates a competitive yet educational atmosphere for its users. Although I did not get to cover other features of the project, I feel that the website is in a healthy state in terms of what its capabilities are.

One of the more valuable lessons from this project was learning from the mistakes I made along the way. Technical challenges encountered along the way pushed the boundaries of my current skills and required me to use innovative solutions and continuous learning, which helped grow my technical skills as a software developer.

In conclusion, CodeHub stands as a demonstration of the educational and developmental journey I had gone through throughout this project. Not only does it fulfil the project’s objective of creating a practical tool for coding practice but it also demonstrates significant personal growth as a developer, aligning with the FYP’s expectation to learn and build upon my mistakes. I feel I can confidently say this project has prepared me well for future professional challenges, as well as installing a deeper understanding of the difficulties of software development.

Reference for Doxygen Location

Regarding the Doxygen Documentation location, you can find the Doxygen index.html file inside the folder with the code submitted in the Final Product, through the file path: **Impact\docs\html\index.html**

Declaration of Plagarism

1. **I know that plagiarism means taking and using the ideas, writings, works or inventions of another as if they were one's own. I know that plagiarism not only includes verbatim copying, but also the extensive use of another person's ideas without proper acknowledgement (which includes the proper use of quotation marks). I know that plagiarism covers this sort of use of material found in textual sources and from the Internet.**

2. **I acknowledge and understand that plagiarism is wrong.**

3. **I understand that my research must be accurately referenced. I have followed the rules and conventions concerning referencing, citation and the use of quotations as set out in the Departmental Guide.**

4. **This assignment is my own work. I acknowledge that copying someone else's assignment, or part of it, is wrong, and that submitting identical work to others constitutes a form of plagiarism.**

5. I have not allowed, nor will I in the future allow, anyone to copy my work with the intention of passing it off as their own work.

Name: SAM CULLEN Student #: C00250093

Signed: Sam Cullen Date: 19/04/2024