COMP 693 Industry Project Final Report

PLANTZZZ UPGRADE: ENHANCING LEARNINGIN PLANT IDENTIFICATION

Independent Project

Submitted By：

Name: Wei (Xiangwei) Zhang
Student ID: 1139462

Date:30/05/2024
Lincoln University

EXECUTIVE SUMMARY

Problem Addressed:
The current Plantzzz App, designed to help users memorize Latin and common plant names, received negative feedback regarding its usability and was built on an outdated platform difficult to update. The School of Landscape Architecture aimed to modernize and rebuild the app.

Goal:
The project's goal was to create an enhanced plant identification learning application with a more dynamic, user-friendly design to improve student learning. The new version sought to provide a modern interface, timely feedback, and features aligned with educational objectives.

Methods:
The project followed an Agile methodology, using wireframes for UI design and developing the frontend with React.js, CSS, Ant Design, and Canvas. The backend was built with Python and Flask, with a MySQL database for data management. Thorough testing, including beta testing and manual unit testing, ensured functionality and usability.

Outcome:
The enhanced Plantzzz App features a modern, user-friendly interface with interactive elements like countdown timers, progress bars, and save progress capabilities. It efficiently manages plant information using a MySQL database. The app's strengths include improved usability and attractive features. Future recommendations include adding real-time collaboration for teachers and students and advanced analytics for learning outcome.

TABLE OF CONTENTS

GLOSSARY/ACRONYMS

- Agile Methodology: An iterative approach to project management and software development.
- AntD (Ant Design): A React UI framework that provides high-quality components for building user interfaces.
- API (Application Programming Interface): A set of tools and protocols that allows different software applications to communicate with each other.
- CORS (Cross-Origin Resource Sharing): A security feature in web browsers that allows or restricts web pages from making requests to a different domain.
- CSS (Cascading Style Sheets): A language used to describe the look and formatting of a website.
- EdTech (Educational Technology): The use of technology to support teaching and learning.
- Flask: A lightweight web application framework for Python.
- GitHub: A platform for version control and collaborative software development using Git.
- MySQL: An open-source relational database management system.
- React.js: A JavaScript library for building user interfaces.
- UI (User Interface): The part of a computer application or device that users interact with.
- UX (User Experience): The overall experience of a person using a product, including how easy and pleasant it is to use.
- Wireframe: A software used to create a basic outline or sketch of a web page or app, used to plan its structure and layout.
- SQL (Structured Query Language): A language used to manage and manipulate databases.
- WSGI (Web Server Gateway Interface): A standard for Python web application development that allows web servers to communicate with web applications.

1. BACKGROUND

1.1 Project Overview

The Enhanced Plant Identification Learning Application (Plantzzz App) is an independent study project offered by the School of Landscape Architecture at the Lincoln University. This project falls under the field of Educational Technology (EdTech). The main focus of this project is to enhance the Plantzzz App, a tool specifically designed to help students learn plant names and identification. The goal is to make learning more interactive and engaging through the use of modern technology. This project aims to provide students with a user-friendly application that supports their learning journey.

1.2 Problem Statement

The current Plantzzz App (hosted at https://plantzzz.web.app/), is an online tool designed to help users differentiate between the Latin and common names of plants. However, users have provided negative feedback about its usability, and it is built on an outdated platform that is difficult to update. The School of Landscape Architecture aims to modernize and rebuild the Plantzzz App through this project.

1.3 People involved in the project

- *Faculty Members (Academic stakeholders)*: The landscape course examiner, Nada and tutor Anna.
- *Developer and UI designer:* Wei (Xiangwei) Zhang manages the app's development and design, ensuring it meets educational goals and provides a user-friendly interface.
- *Beta Tester*: Dian Gao, the student of Applied computing, was invited to be the beta tester of the Plantzzz Application.

2. REQUIREMENTS AND GOALS

2.1 Overall Goal

The overall goal of this project is to design and develop a plant recognition learning application. The app aims to use a more dynamic and user-friendly design to enable students to more effectively memorize and identify plant names. The new version of the app will provide a more user-friendly interface, deliver timely feedback to students as well as align with the educational objectives set by the course instructors.

2.2 Specific Objectives

- Design a Scalable Database
- Create a User-Friendly Platform
- Improve student and teacher dashboards and other functional features

2.3 Success Criteria

The success of the project will be measured by the following criteria:

- Database Implementation: A database that efficiently manages plant information.
  How to evaluate: The database can store all plants' name information provided by the Landscape tutors and can be effectively used in web App.
- User-Friendly Interface: A modern UI design that allows users to navigate the app easily and interact with its features smoothly.
  How to evaluate: Evaluation on the usability obtained from the beta tester. If the evaluation is that it is easy to operate, the design of this part meets the requirements.
- Functional Features: The app can support students in effectively identifying plant name knowledge across all four semesters. All designed features, such as quizzes and dashboards, should work well, providing an engaging learning experience without issues.

2.4 Literature Review

In the field of educational technology, gamification has become a key strategy to improve student engagement and learning outcomes. Apps like Kahoot! and Quizizz exemplify this approach by incorporating game-like elements into the educational process.
Kahoot!, is designed to foster student engagement and motivation through its interactive setup, utilizing features such as real-time feedback, competitive games, and more. The effect of increasing student engagement and perceived learning outcomes has been well documented through extensive research (Wang & Tahir, 2020).
Likewise, Quizizz provides a gamified learning experience that enhances classroom activities by making them more

interactive and reducing distractions. It stands out for its rich features, including a game-like environment with levels, leaderboards and a user-friendly interface (Meng et al., 2019).

3. METHOD

3.1 Overview

The project followed an Agile methodology, using word documents to record weekly tasks and priorities. For UI design, wireframes were created to plan the layout. A MySQL database was set up to store data. Frontend development was carried out using React.js, CSS, and front-end libraries like Ant Design (AntD) and Canvas, focusing on developing UI components and implementing data fetching and rendering. The backend was developed with Python and Flask, featuring APIs for communication, Flask Blueprints for modularity, and CORS for cross-origin requests. Version control was managed on GitHub with regular code commits to maintain version history.

3.2 Design

Software Components:

Login/Register
It includes separate registration pages for both students and teachers, along with a login page for all users.

My Profile

This component enables students/teachers to view and edit their information.

Student Dashboard

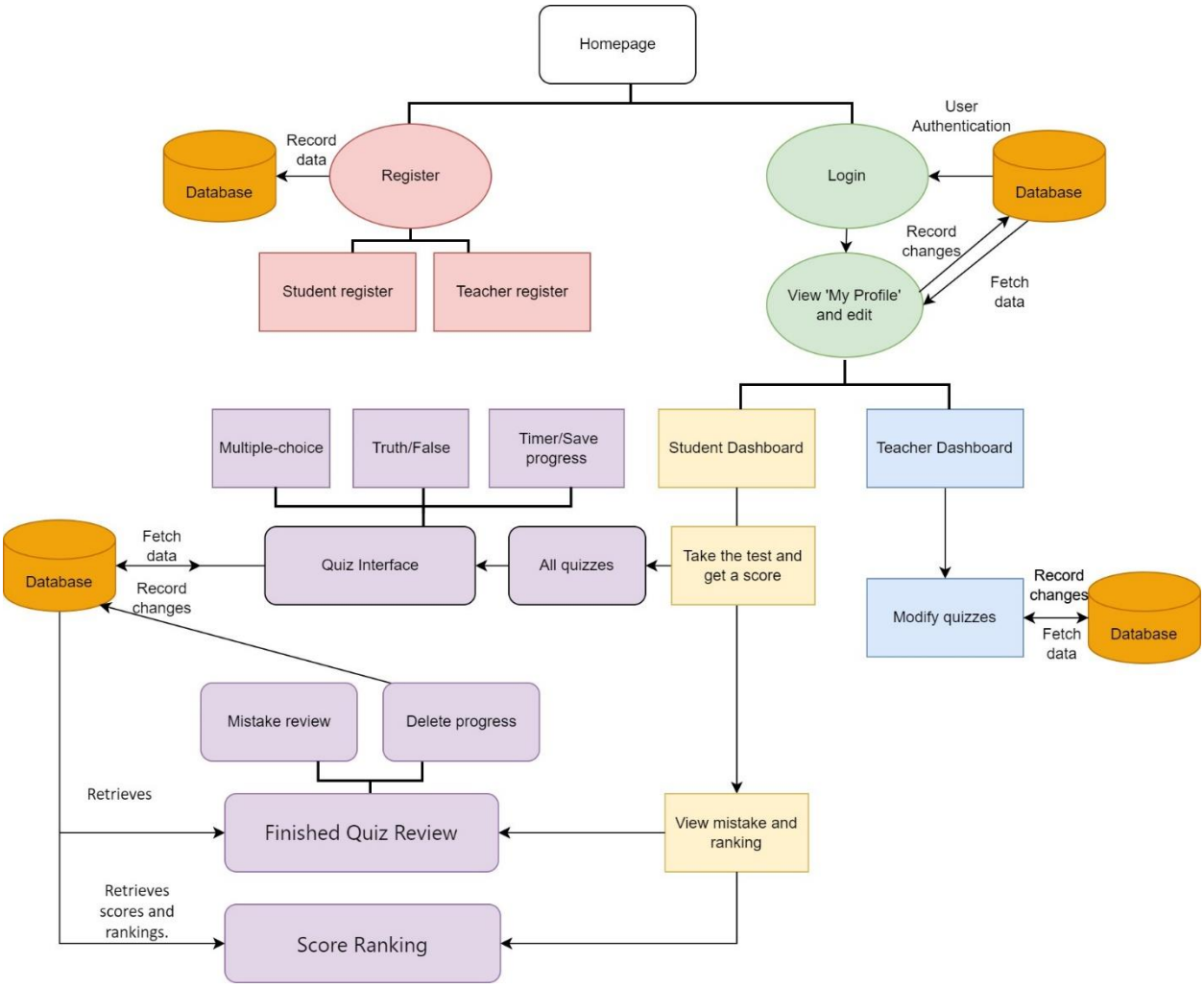
1. *All Quizzes*: Students have access to quizzes from various semesters and can access them by clicking a designated icon.
2. *Quiz Interface*: It contains different question types, such as true/false and multiple-choice questions, as well as a progress bar, question timer and the function of saving progress.
3. *Finished quiz review:* Allows students to review their completed quizzes, delete records, view summarized incorrect answers, and learn from mistakes by linking plant names to Wikipedia.
4. *Score Ranking*: The leaderboard feature enables students to see their rankings based on quiz scores, promoting a competitive learning atmosphere

Teacher Dashboard:

1. *Edit Quiz*: This component enables teachers to alter quiz questions, related photos, and correct options.

Figure 1.

Components and interactions of Plantzzz Application. Own creation.



Component Interactions with Database:

Login/Register:

- *User Authentication*: Verifies user credentials against the database.
- *Record Data*: Stores student and teacher registration data.

My Profile:

- Fetches and updates user information.

Student Dashboard:

- *All Quizzes*: Loads available quizzes.
- *Quiz Interface*: Loads quiz content and saves progress.
- *Finished Quiz Review*: Retrieves and updates completed quiz data.

- *Score Ranking*: Retrieves scores and rankings.

Teacher Dashboard:

- *Edit Quiz*: Loads and saves quiz modifications.

3.3 Risks and Challenges

Table 1.

Risk, strategy and impact on project, own creation.

Risk/Challenge	Strategy	Impact on Project
Due to limited demand collection methods, it may not be possible to fully understand the functions that users need, making it difficult to attract users.	Developed an early prototype; prepared detailed questions for stakeholder meetings.	Enhanced stakeholder engagement, ensuring app met usability standards.
Lack of real student testing might lead to missed usability issues.	Invite a Lincoln student unrelated to the project to take a test and get feedback on her learning process	Increased confidence in the app's functionality and user interface.
Unfamiliarity with React and JavaScript lengthening the learning curve, impacting project timeline.	Use online resources to study and stick to the plan	Minimize development delays.

3.4 Implementation

Step 1: Requirements gathering and UI Design

Created initial wireframes for the user interface layout, present to Landscape School lecturers at conferences and get advice.

Step 2: Database Setup:

1. Built a MySQL database to store plant data, including images, Latin names, and common names.
2. For quiz information, created tables to store semester details, quiz questions and answers, as well as question types (true/false, multiple choice).
3. For users, developed tables to store user information and user types (student or teacher).
4. For user progress, established tables to record start and end times, scores, and answers for each question.
5. Import data from the existing Excel and generate SQL insert statements.

Step 3: Backend Development:

1. Used Flask Blueprints for modular development; created user.py, student.py, and teacher.py modules.
2. Implemented CORS to allow the frontend and backend to communicate when running on different localhost ports.
3. Developed APIs with Python and Flask for user authentication, data retrieval, and updates.

Step 4: Frontend Development:

1. Built the frontend using React.js, developing components for navigation, login/register, student dashboard, teacher dashboard, and quiz interface etc.
2. Implemented data fetching and rendering from the backend APIs.
3. Enhanced the UI with CSS for styling and Canvas, AntD for interactive elements.

Step 5: Integration:

1. Integrated the frontend with the backend APIs to enable communication and data flow.
2. Ensured that user actions on the frontend triggered appropriate backend responses and updates.

Step 6: Testing:

1. Conducted thorough testing of both frontend and backend components to ensure functionality and usability.
2. Invited a Lincoln student unrelated to the project to test the application and provide feedback on the learning process.
3. Improved the application based on test results and user feedback.

Others: Version Control and Documentation:

1. Maintained comprehensive project documentation, including a README and project reports (weekly, mid-term, and final summary reports). PowerPoint presentation summarizing the project.
2. Regularly committed code to GitHub to keep track of version history and project progress.

Artifacts Produced:

The project produced several key artifacts, including the source code hosted on GitHub, a README document, a detailed project report, wireframe diagrams of the UI design, and a PowerPoint presentation summarizing the project.

4. RESULTS AND OUTCOMES

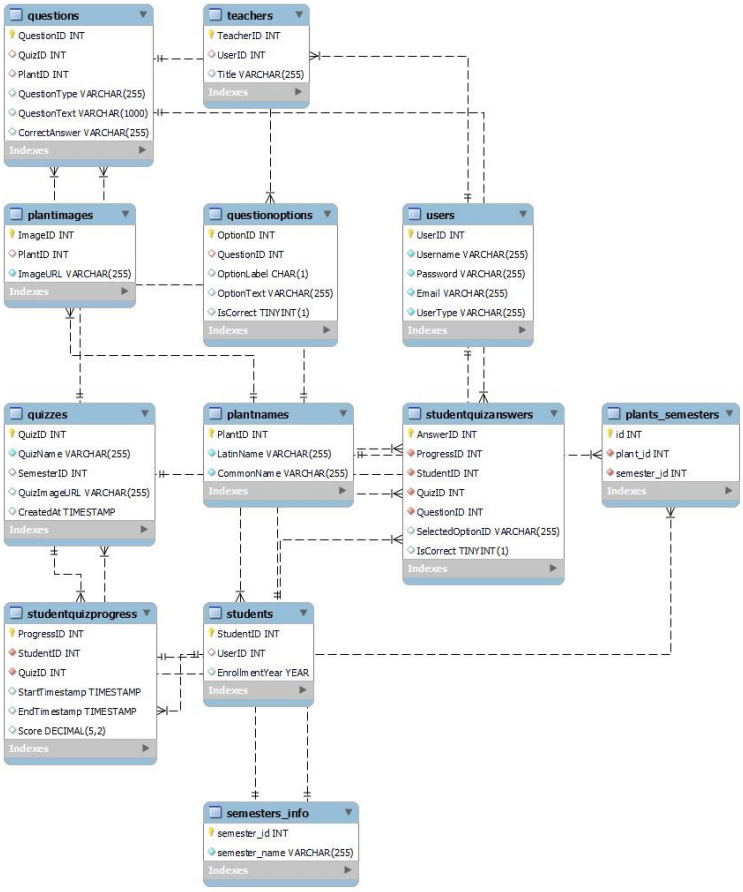
4.1 Evidence of Deliverables

Database Implementation:

Evidence 1: Picture of the MySQL database schema, including tables for plant data, quizzes, user information, and progress etc. The schema includes several tables with relationships that support the application's functionality.

Figure 2.

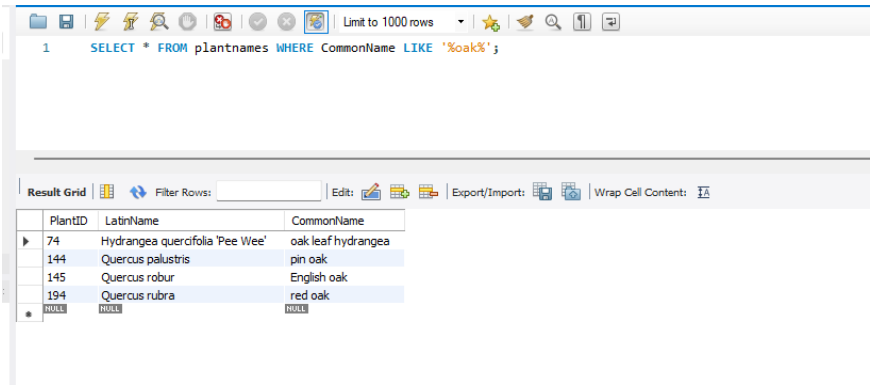
Plantzzz Application Database Schema, own creation.



Evidence 2: Example Queries and Results the database efficiently stores and manages all the plant information provided by the Landscape tutor.

Figure 3.

Plantzzz Application Database Example Queries and Results, own creation.



Evidence 3: Supporting Documentation: SQL scripts used for database creation. See in Appendix A. SQL script screenshot.

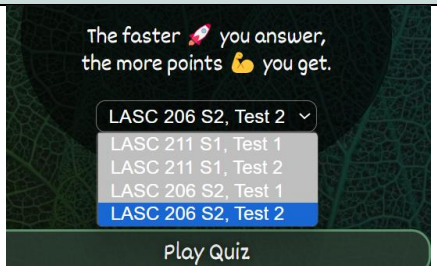
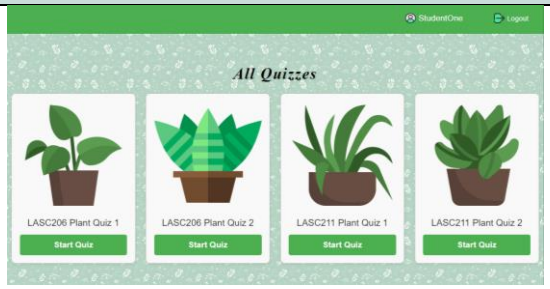
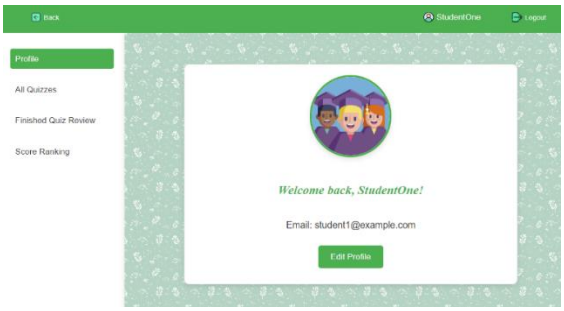
User-Friendly Interface:

Evidence 1: Screenshots of quiz interface and student dashboard upgrade

Added a new dashboard for easier navigation, allowing students to quickly access their profile, quizzes, finished quiz reviews, and score rankings. The old version did not have a dashboard. Quiz interface enhanced with plant icons, making it more attractive to users.

Table 2.

Plantzzz Application User Interface comparison (old version vs. enhanced application).

	Old version Plantzzz	Enhanced Plantzzz
Quiz interface	 Old version Plantzzz App (https://plantzzz.web.app/)	
Student dashboard	Old version does not have student dashboard	

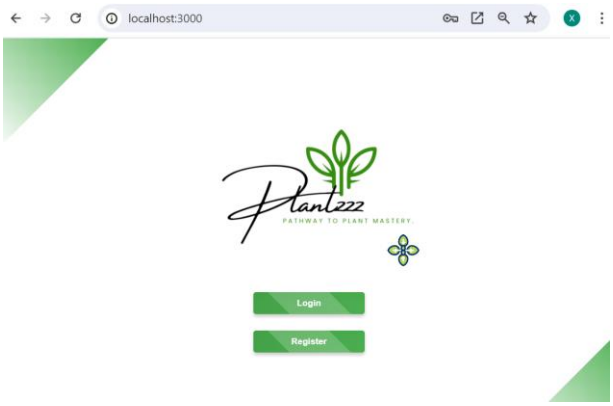
Evidence 2: Wireframes used during the planning phase. The screenshot is a first draft of the design, which was used to discuss user needs with stakeholders.

Figure 4.

Initial wireframe, own creation.

Figure 5.

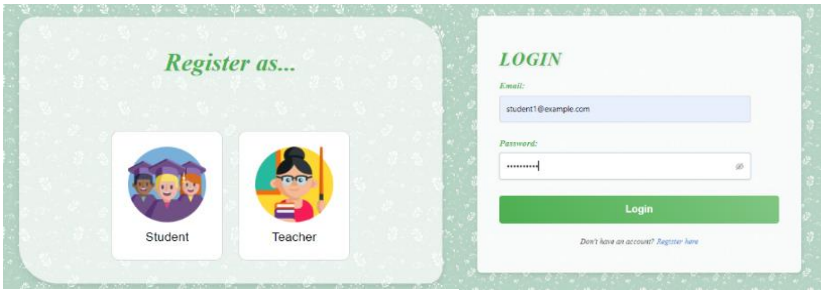
Final UI screenshot, own creation.

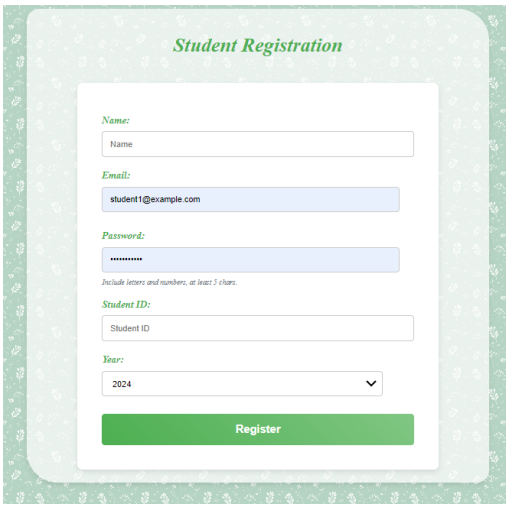


Functional Features:

Evidence 1: Users login and register

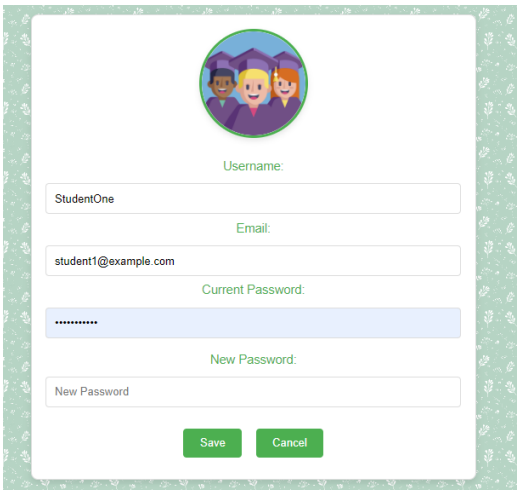
Figure 6. Users' login and register, own creation.





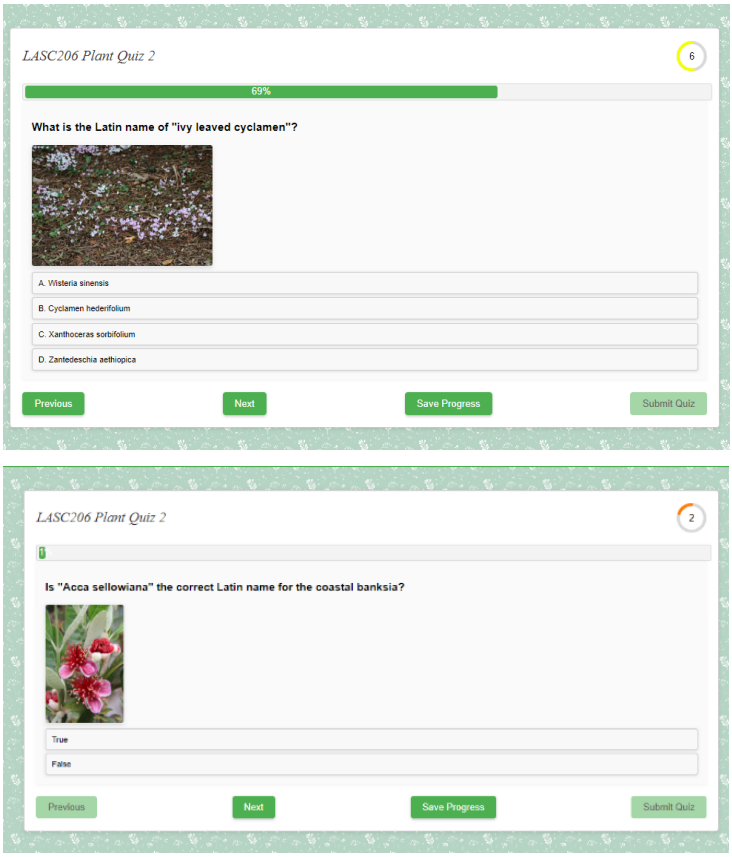
Evidence 2: Users edit their profile

Figure 7. Users edit their profile, own creation.



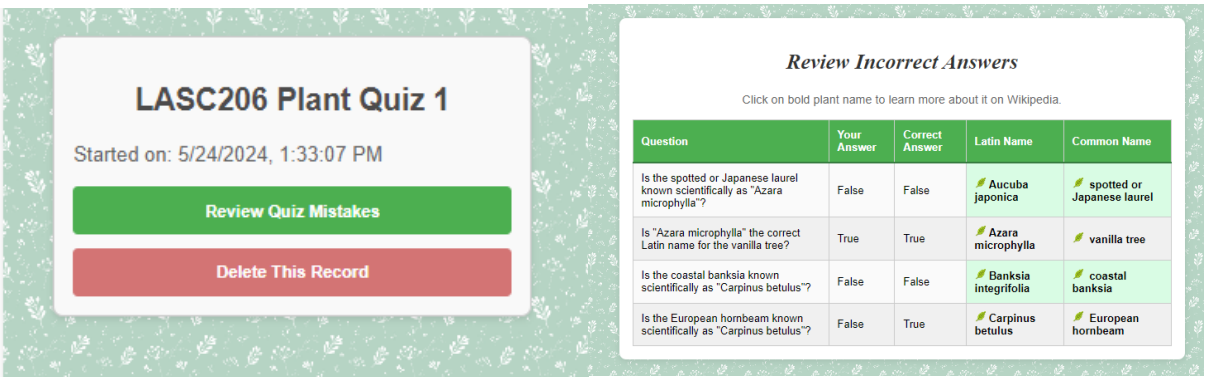
Evidence 3: Students accessing quizzes and viewing quiz content.

Figure 8. Students accessing quizzes and viewing quiz content (Multi-choice, Truth/False question types), own creation.



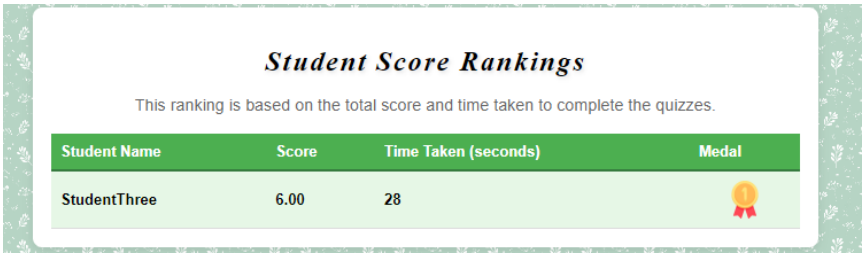
Evidence 4: Students reviewing their completed quizzes and viewing mistake summaries.

Figure 9. Students reviewing their completed quizzes records, delete records and view mistake summaries, own creation.



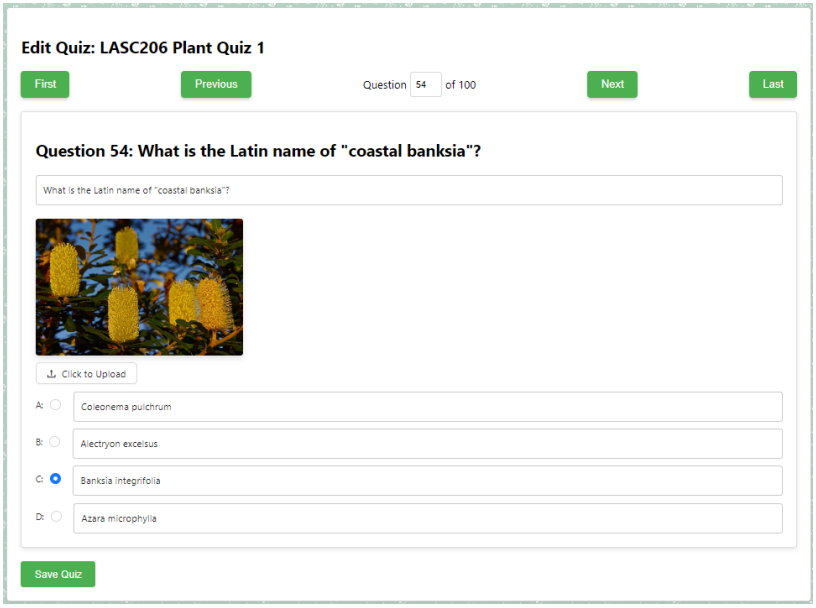
Evidence 5: Students reviewing their score rankings.

Figure 10. Students reviewing their score rankings, own creation.



Evidence 6: Teachers modifying quiz questions

Figure 11. Teachers modifying quiz questions, own creation.



4.2 Testing/Validation

To ensure the Plantzzz App met all specified requirements, I conducted several stages of testing, including beta testing, unit testing, and end-to-end testing.

Beta Testing

Method:

Conducted face-to-face interviews with a beta tester, Dian, an applied computing student from Lincoln University who is not related to this project. Dian suggested adding more interactive features, such as linking incorrect answers to Wikipedia for additional learning, implementing a countdown timer, and adding password visibility toggles. Based on this feedback, several optimizations were made, including these enhancements and improving the overall responsiveness of the UI.

Evidence:

Figure 12.1 Beta tester Feedback email 1

Re: Feedback on the Plantzzz App Testing Session 收件箱 ×

DIAN GAO
发送至 我

20:32 (0分钟前)

Dear Wei,

Thank you for inviting me to participate in the testing session of the Plantzzz App. It was enlightening to experience the application firsthand.

The app is impressively designed. However, I have a few suggestions that could enhance its functionality and user experience:

1. **Navigation for Incorrect Answers:** It would be beneficial to include an instruction for the feature that navigates students to the Wikipedia page of the plant names they answered incorrectly in the quiz. This could make them feel easier to understand and use the function.
2. **Countdown Timer for Quizzes:** Instead of a standard clock counting up the time, a countdown timer for each question might be more effective. This could help students better manage their time during the quizzes, making the testing experience more engaging and less stressful.
3. **Password Visibility Toggle on Login Page:** Adding a toggle to make the password visible on the login page could improve user experience.

I hope these suggestions prove useful in refining the app further. Wishing you a productive and enjoyable weekend!

Cheers,

Figure 12.2 User Interface adjustments based on feedback 1 and 2

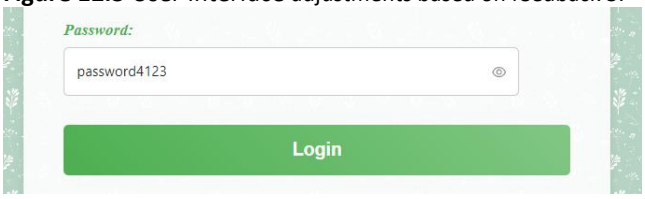
Click on bold plant name to learn more about it on Wikipedia.

Question	Your Answer	Correct Answer	Latin Name	Common Name
Is the spotted or Japanese laurel known scientifically as "Azara microphylla"?	False	False	**Aucuba japonica**	**spotted or Japanese laurel**

LASC206 Plant Quiz 2 6

69%

Figure 12.3 User Interface adjustments based on feedback 3.

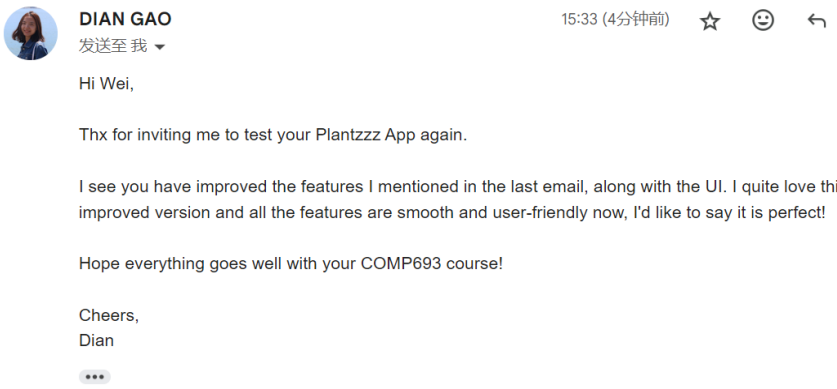


Password:

password4123

Login

Figure 12.4 Beta tester Feedback email 2



DIAN GAO
发送至 我
15:33 (4分钟前)

Hi Wei,

Thx for inviting me to test your Plantzzz App again.

I see you have improved the features I mentioned in the last email, along with the UI. I quite love this improved version and all the features are smooth and user-friendly now, I'd like to say it is perfect!

Hope everything goes well with your COMP693 course!

Cheers,
Dian

Unit Testing

I conducted manual unit testing for the entire Plantzzz application by identifying and testing its various components and features. This included user registration, login, dashboard, quiz interface, edit quiz, score ranking, mistake summary, and data interactions with the database. Below is an example of testing the quiz interface, illustrating how I tested the quiz interface, my testing process, and the related evidence. Other components followed the same process.

Step 1. Identify the Components to test

Test the quiz interface component, focusing on functionalities such as loading quiz data, displaying questions, handling option selection, saving progress, and submitting quizzes.

Step 2. Create Test Cases:

- Test Case 1: Load quiz data

Input: Open the quiz page and select quiz to start.
Expected Outcome: Quiz details and the first question are loaded and displayed correctly.

- Test Case 2: Select an option

Input: Click on an option for the current question.
Expected Outcome: The selected option is highlighted, and feedback is displayed ("Correct Answer!" or "Wrong Answer!").

- Test Case 3: Navigate between questions

Input: Click the "Next" button to go to the next question.
Expected Outcome: The next question is displayed, and the progress bar updates.

- Test Case 4: Save progress

Input: Click the "Save Progress" button.
Expected Outcome: Progress is saved, and a confirmation message is displayed.

- Test Case 5: Submit quiz

Input: Click the "Submit Quiz" button after saving progress.
Expected Outcome: Quiz is submitted, the final score is displayed, and the "Submit Quiz" button is disabled.

Step 3. Execute Tests Manually

Note: Appendix B Include screenshots of each step and the resulting UI state.

Table 3. Test Case and outcome summary for quiz interface.

Test Case	Input	Expected Outcome	Actual Outcome	Pass/Fail
TC1	Open quiz page with valid quiz ID	Quiz details and first question are loaded correctly	Quiz details and first question loaded correctly	Pass

TC2	Click on an option	Option is highlighted, feedback is displayed	Option highlighted; feedback displayed	Pass
TC3	Click "Next" button	Next question displayed, progress bar updates	Next question displayed; progress bar updated	Pass
TC4	Click "Save Progress" button	Progress saved; confirmation message displayed	Progress saved; confirmation message displayed	Pass
TC5	Click "Submit Quiz" button after saving	Quiz submitted, final score displayed, button disabled	Quiz submitted, final score displayed, button disabled	Pass

5. REFLECTIONS

5.1 Reflections

Despite initial challenges and a slow start, I achieved the primary goals, including designing a database, creating a user-friendly interface, and adding new interactive features. Some tasks took longer than expected, but as the project progressed, my task management and efficiency improved significantly.

To complete the project, I had to learn several new skills. I developed dynamic UI components using React.js, handled backend development with Python and Flask, and designed a MySQL database for storing plant and quiz data. Additionally, I enhanced the UI using CSS and libraries like Ant Design (AntD) and Canvas.

Through this project, I grew both personally and technically. I improved my problem-solving and time management skills, became more adaptable to changes, and learned the importance of continuous learning. Overall, I am proud of the final product and my progress throughout the semester.

5.2 Conclusions

Strengths and limitations of the Plantzzz project

The strengths of the Planzzz app include a modern, user-friendly interface that enhances the learning experience with interactive features such as a countdown timer, progress bar, and save progress functionality. The use of React.js and Flask provides a robust and responsive platform, and the MySQL database ensures efficient data management.

However, there are limitations. The app lacks real-time interaction features between students and teachers and advanced analytics for learning outcomes. In addition, due to time constraints, more interesting and engaging interactive content was not developed.

Future recommendations and lessons learned

For future improvements, I would recommend adding real-time collaboration features, advanced analytics for learning outcomes, and an offline mode for improved accessibility. If I were to do the project again, I would focus on better initial planning and incorporate automated testing from the beginning to improve efficiency and quality.

6. REFERENCES

Meng, C. K., Ming, T. M., Nasir, J. S. M., & Koo, A. C. (2019). A Gamified Classroom with Technical and Vocational Education and Training (TVET) Students using Quizziz. *ResearchGate*.https://www.researchgate.net/publication/336513666_A_Gamified_Classroom_with_Technical_and_Vocational_Education_and_Training_TVET_Students_using_Quizziz

Wang, A. I., & Tahir, R. (2020). The effect of using Kahoot! for learning – A literature review. *Computers & Education*, *149*, 103818. https://doi.org/10.1016/j.compedu.2020.103818

7. APPENDICES

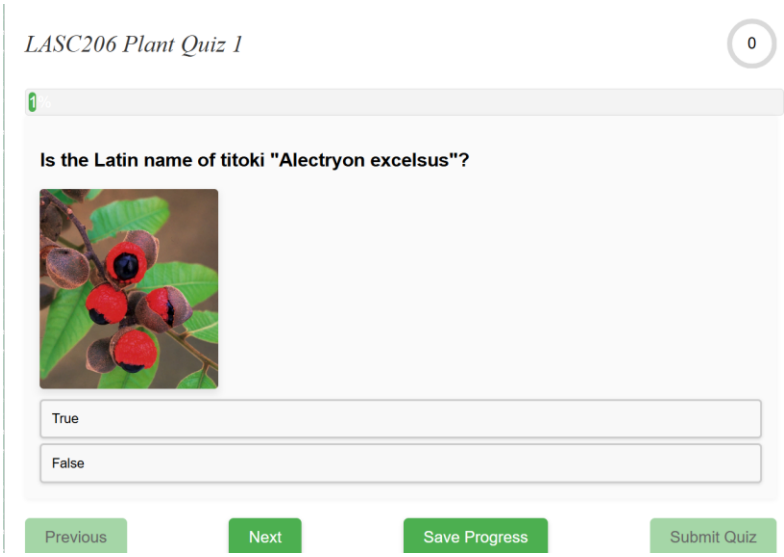
Appendix A. SQL script screenshot.

```
DROP SCHEMA IF EXISTS plantzzz;
CREATE SCHEMA plantzzz;
USE plantzzz;


CREATE TABLE Users (
    UserID INT AUTO_INCREMENT PRIMARY KEY,
    Username VARCHAR(255) NOT NULL,
    Password VARCHAR(255) NOT NULL,
    Email VARCHAR(255) NOT NULL,
    UserType VARCHAR(255) NOT NULL
);

CREATE TABLE Students (
    StudentID INT PRIMARY KEY,
    UserID INT,
    EnrollmentYear YEAR,
    FOREIGN KEY (UserID) REFERENCES Users(UserID) ON DELETE CASCADE
);

CREATE TABLE Teachers (
    TeacherID INT PRIMARY KEY,
    UserID INT,
    Title VARCHAR(255),
    FOREIGN KEY (UserID) REFERENCES Users(UserID) ON DELETE CASCADE
);


CREATE TABLE PlantNames (
    PlantID INT AUTO_INCREMENT PRIMARY KEY,
    LatinName VARCHAR(255) NOT NULL,
    CommonName VARCHAR(255) NOT NULL
);
```

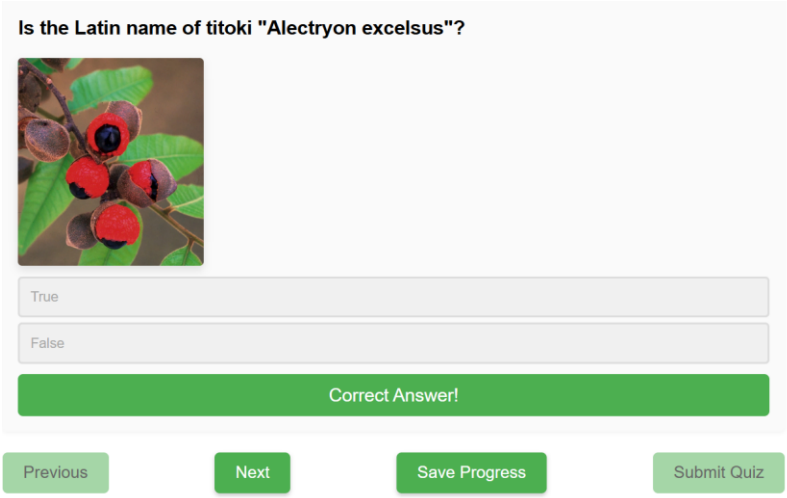
```sql
CREATE TABLE semesters_info (
  semester_id INT AUTO_INCREMENT PRIMARY KEY,
  semester_name VARCHAR(255) NOT NULL
);

CREATE TABLE plants_semesters (
  id INT AUTO_INCREMENT PRIMARY KEY,
  plant_id INT NOT NULL,
  semester_id INT NOT NULL,
  FOREIGN KEY (plant_id) REFERENCES PlantNames(PlantID),
  FOREIGN KEY (semester_id) REFERENCES semesters_info(semester_id)
);

CREATE TABLE PlantImages (
    ImageID INT AUTO_INCREMENT PRIMARY KEY,
    PlantID INT,
    ImageURL VARCHAR(255) NOT NULL,
    FOREIGN KEY (PlantID) REFERENCES PlantNames(PlantID) ON DELETE CASCADE
);

CREATE TABLE Quizzes (
    QuizID INT AUTO_INCREMENT PRIMARY KEY,
    QuizName VARCHAR(255) NOT NULL,
    SemesterID INT,
    QuizImageURL VARCHAR(255),
    CreatedAt TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (SemesterID) REFERENCES semesters_info(semester_id)
);

CREATE TABLE Questions (
    QuestionID INT AUTO_INCREMENT PRIMARY KEY,
    QuizID INT,
    PlantID INT,
    QuestionType VARCHAR(255),
    QuestionText VARCHAR(1000),
    CorrectAnswer VARCHAR(255),
    FOREIGN KEY (QuizID) REFERENCES Quizzes(QuizID) ON DELETE CASCADE,
    FOREIGN KEY (PlantID) REFERENCES PlantNames(PlantID) ON DELETE CASCADE
);

CREATE TABLE QuestionOptions (
    OptionID INT AUTO_INCREMENT PRIMARY KEY,
    QuestionID INT,
    OptionLabel CHAR(1),
    OptionText VARCHAR(255),
    IsCorrect BOOLEAN DEFAULT FALSE,
    FOREIGN KEY (QuestionID) REFERENCES Questions(QuestionID) ON DELETE CASCADE
);

CREATE TABLE StudentQuizProgress (
    ProgressID INT AUTO_INCREMENT PRIMARY KEY,
    StudentID INT NOT NULL,
    QuizID INT NOT NULL,
    StartTimestamp TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    EndTimestamp TIMESTAMP,
    Score DECIMAL(5,2),
    FOREIGN KEY (StudentID) REFERENCES Students(StudentID) ON DELETE CASCADE,
    FOREIGN KEY (QuizID) REFERENCES Quizzes(QuizID) ON DELETE CASCADE
);

CREATE TABLE StudentQuizAnswers (
    AnswerID INT AUTO_INCREMENT PRIMARY KEY,
    ProgressID INT NOT NULL,
    StudentID INT NOT NULL,
    QuizID INT NOT NULL,
    QuestionID INT NOT NULL,
    SelectedOptionID VARCHAR(255),
```

Appendix B. screenshots of each step and the resulting UI state

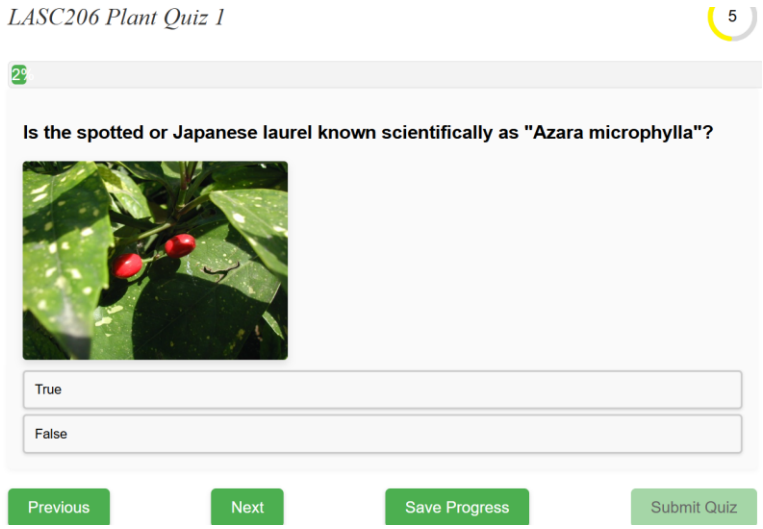
Quiz details and first question loaded correctly



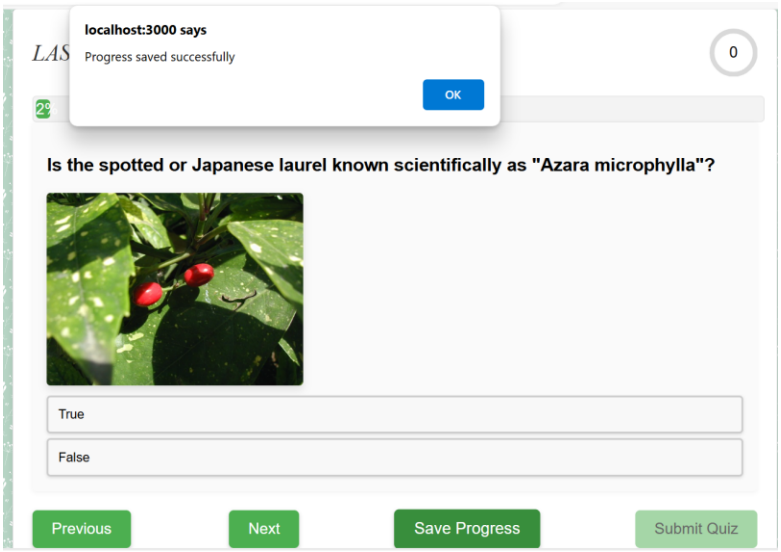
Option highlighted, feedback displayed



Next question displayed, progress bar updated



Progress saved, confirmation message displayed



Quiz submitted, final score displayed, button disabled