

SCHOOL OF COMPUTING

DIPLOMA IN INFORMATION TECHNOLOGY
DIPLOMA IN APPLIED AI AND ANALYTICS

ST0503 BACKEND WEB DEVELOPMENT

2023/2024 SEMESTER 2
PRACTICAL ASSIGNMENT 2

Objective of Assignment

To allow students to practice what they have learnt in the module by creating a task tracker website to gamify the sustainability movement. The goal is to encourage users to complete environmentally-friendly tasks and earn rewards while tracking their progress.

Instructions and Guidelines:

The assignment should be done individually and will account for **30%** of your final grade.

1. The assignment should be submitted by **Friday, 2 Feb, 2024 11:59 pm**.
2. You are **REQUIRED** to do the following for your submission
 - Use the **Github Classroom repository** provided to commit all your progress and changes. **It will be reviewed for marking**.
 - **Complete the CA2 Individual Report**. The form is available in BrightSpace under Assignments > Forms > CA2 Individual Report.
 - Prepare the **Declaration of Academic Integrity (SOC) form**. **Your Assignment will NOT be marked if the form is not submitted**. The form is available in BrightSpace under Assignments > Forms > Declaration of Academic Integrity (SOC) form.

- You are also required to **zip up your source code** along and submit it along with your files to Brightspace. You are to remove the **node_module** folder when zipping the files.
- Below is example of what your submission should contain

<ul style="list-style-type: none">— BED-CA2-PYYYYYY-CLASS.zip— Declaration of Academic Integrity (SOC).docx— CA2 Individual Report.docx

3. You are required to develop a Frontend Website for Backend Server in NodeJS with MySQL Database developed. You are to enhance your Backend Server with security features such as authentication and password hashing.
4. The interview will be conducted during the practical lessons on week 17. You are expected to explain the program logic and modify the program during the interview. If you are absent from the interview, **you will be awarded zero mark for the assignment.**
5. **No marks will be awarded**, if the work is copied or you have allowed others to copy your work.
6. **50%** of the marks will be deducted for assignments that are received within **ONE (1) calendar** day after the submission deadline. No marks will be given thereafter.



Warning: Plagiarism means passing off as one's own the ideas, works, writings, etc., which belong to another person. In accordance with this definition, you are committing plagiarism if you copy the work of another person and turning it in as your own, even if you would have the permission of that person.

Plagiarism is a serious offence, and if you are found to have **committed, aided, and/or abetted** the offence of plagiarism, disciplinary action will be taken against you. If you are guilty of plagiarism, you may **fail all modules** in the semester, or even be **liable for expulsion**.

Task Tracker Backend Server with Frontend Integration

Section A: Frontend and User Authentication

In this project, you will be focusing on the frontend development of your task tracker, which you designed and built in CA1. Your task is to create an interactive and user-friendly interface that seamlessly integrates with the backend, allowing users to engage with tasks, track progress, and interact with the gamified elements you've designed.

Requirements:

1. User Registration and Login Pages:

Design and implement user registration and login pages that provide a seamless entry point to your application. Utilize JWT for secure authentication.

2. Authentication Flow:

Develop a smooth authentication flow that grants access to authorized users while keeping their data secure. Tokens should be managed effectively.

3. Password Security:

Implement Bcrypt to securely hash and store user passwords on the backend, ensuring a high level of password protection.

4. Messaging:

Create a global page where users can write global messages, and edit or delete their own message. Each user can view all messages written by users. **Source code example of the backend for this requirement is provided in Brightspace under Assignments > CA2 > Messaging Example**

5. Bootstrap Integration:

Apply Bootstrap to enhance the visual appeal and responsiveness of your frontend components, maintaining a consistent and user-friendly design.

6. Endpoint Integration:

Utilize the Fetch API, Axios, or jQuery to connect your frontend with the backend endpoints you've designed in CA1. Ensure seamless communication for task retrieval and completion.

Section B: Extending Functionality

In this section, you will enhance the functionality of your frontend by integrating additional endpoints and refining the user interface.

Requirements:

1. **Integrate Additional Endpoints:**

Extend your frontend to incorporate the additional endpoints you designed in CA1. This could encompass features you have designed in CA1 such as quests, rewards, inventory, etc.

2. **User Interaction Enhancement:**

Develop interactive frontend elements that allow users to manage features you have designed in CA1 such as quests, rewards, inventory with ease, enriching their engagement.

3. **Backend Enhancement:**

Enhance your backend to incorporate the additional endpoints and features. You will need to implement frontend integration for these new features.

Assessment Requirements

We will be evaluating different aspects of your gamified sustainability task tracker website. These aspects, or competencies, are critical to the overall success and effectiveness of the application. Please take this as an opportunity to showcase your skills and creativity in designing and implementing a gamified platform that promotes sustainability and eco-friendly behaviours.

Below are the key competencies that will be assessed:

1. **Backend Server** (10 Marks)

Your backend server's implementation will be evaluated for its functionality and responsiveness. We'll assess how well it handles incoming requests, communicates with the database, and provides appropriate responses.

2. **Functionality** (10 Marks)

This competency assesses how well your task tracker's core features are implemented. We'll evaluate whether users can seamlessly register, log in, complete tasks, earn rewards, and engage with the messaging system.

3. **Code Quality** (15 Marks)

We will review your codebase for readability, organization, and adherence to coding best practices. Clear and well-structured code enhances collaboration and maintainability.

4. **Modularity** (5 Marks)

Modularity involves breaking down your code into reusable modules. We'll assess how well you've organized your frontend and backend code to promote reusability and maintainability.

5. **Error Handling** (5 Marks)

Effective error handling ensures a smooth user experience. We'll examine how well you handle errors, provide informative error messages, and gracefully manage exceptional scenarios.

6. **Documentation** (5 Marks)

Documentation is crucial for others to understand and work with your code. We'll assess the clarity and completeness of your inline comments and external README file.

7. **JWT for Authentication** (10 Marks)

Your implementation of JWT-based authentication will be evaluated for security and reliability. We'll examine how well you generate, validate, and manage user tokens.

8. **BCrypt for Password Hashing** (10 Marks)

Secure password hashing is essential for protecting user data. We'll review how you implement Bcrypt to securely store and validate user passwords.

9. **Frontend to Backend HTTP Requests** (10 Marks)

This competency assesses your frontend's ability to communicate with the backend through HTTP requests. We'll evaluate the accuracy of data retrieval and submission.

10. **HTML & CSS Proficiency** (5 Marks)

Effective use of HTML and CSS enhances user interface design. We'll assess your proficiency in creating structured layouts and visually appealing styles.

11. **Frontend Framework** (5 Marks)

If you've integrated a frontend framework, we'll evaluate its implementation and how well it enhances the user experience and simplifies complex tasks.

12. **JavaScript and DOM Manipulation** (10 Marks)

Skillful use of JavaScript for dynamic content and DOM manipulation will be assessed. We'll examine how well your frontend interacts with users and updates content.

Please approach each competency with creativity and attention to detail. The focus is on understanding your approach, decision-making, and implementation choices throughout the project.

Version Control Requirement

In this project, we will be using Git for version control and GitHub Classroom for managing code submissions. Version control is a crucial aspect of software development that allows you to track changes, collaborate effectively, and maintain a clean and organized codebase. To ensure smooth progress and efficient evaluation, it is mandatory for all students to commit their code regularly using Git.

Version Control with Git:

1. **Commit Regularly:** It is essential to commit your code regularly (**minimally once a week**) as you make progress on your project. Regular commits demonstrate a well-organized development process, help track changes, and make it easier to revert to previous states if needed.
2. **Commit Rationale:** For each commit, provide a clear and concise rationale in the commit message. A good commit message explains the changes made in the commit and why those changes were made. This helps reviewers understand your thought process and the purpose of each commit.

GitHub Classroom Submissions:

1. **Use GitHub Classroom:** All code submissions must be made through GitHub Classroom. Follow the provided repository link to create your personal repository and commit your code into it.
2. **Submission README:** Include a detailed README file in your repository. The README should contain clear instructions on how to run your application, any prerequisites or dependencies, and any additional details that will assist the reviewers in evaluating your project.

Penalties for Non-Compliance:

Failure to adhere to the version control requirements and submission guidelines may result in penalties that could impact your final evaluation. Penalties may include:

- **Disorganized Commits:** Poorly organized commits without clear rationale may lead to a deduction of points in the code quality assessment.
- **Incomplete Submission:** If your submission is incomplete or missing critical components, it may be deemed ineligible for evaluation, resulting in a zero score for the affected parts.

Note: The goal of these requirements is to promote good development practices and ensure a fair and transparent evaluation process. By adhering to the version control guidelines and submission protocols, you'll have a well-documented, organized, and properly managed project.

-- End of Assignment Brief --