Bryce Bien

Team Member Information

- Email: **bienb1@mymail.nku.edu (mailto:bienb1@mymail.nku.edu)**
- GitHub: **https://github.com/brycebien/ASE-285-Team-Project (https://github.com/brycebien/ASE-285-Team-Project)**

Project Plan

Feature 1: Subtasks

Rationale

Subtasks in a to-do list help users break down larger tasks into smaller tasks. This makes tackling larger tasks easier allowing the user to provide smaller tasks within the large one. The technology required includes database for storing the subtask under the large task, Node.js for backend logic, HTML, CSS, and EJS for frontend. To implement the task, the schema and UI need to be modified.

Requirements

As a student, I want to create smaller tasks that need to be completed for a large project so that I can stay on track while working on it.

- As a user, I want to click a button next to my task to add a subtask so I can stay on track with the large task.
- As a user, I want to create subtasks for my main task so I can break down larger projects into manageable steps.
- As a user, I want to mark subtasks as complete individually so I can track my progress.
- As a user, I want to see all subtasks associated with a main task so I can see the bigger picture and plan my work accordingly.
- As a user, I want to see the progress of subtasks within a main task so I can quickly assess how close I am to completing the entire task.

Deliverables

- Design:
  - GitHub will be linked
- code:
  - *GitHub will be linked*
- test:

    - *GitHub will be linked*
- documents:
    - *GitHub will be linked*

Test Plan

Subtask Tests:

Automated

- Test that adding a subtask returns the expected results.
    - Verify that the task is stored properly in the database
- Test that write and update works successfully

Manual

- Test that the user is able to input and submit a subtask

Feature 2: Password Encryption and User Data Storage

Rationale

Password encryption and user data storage are required in any software application to ensure the user's information is protected. For this feature, we will need an encryption method, database, and Node.js for backend logic. To implement this feature the backend must be modified to encrypt the user's password information and store it securely into the database.

Requirements

As a student, I want my user information to be secure so I can not worry about my information getting stolen in the event of a data breach.

- As a user, I want my password to be securely transmitted to the server when logging in so I can trust my login credentials are protected.
- As a user, I want my password to be encrypted on the client-side before transmission so I can have confidence that my password remains confidential even if intercepted by attackers.

Deliverables

- Design:

  Github will be linked

- code:

  Github will be linked

- test:

Github will be linked

- documents:

  Github will be linked

Test Plan

Password Security Tests:

Automated:

- Test successful encryption of password credentials
- Test successful encrypted storage of password credentials in database

Milestones and Deadline

- Milestone 1: Initial Prototype and Canvas Page done.
  - Date: 3/24/24
  - Goal: By the above date, the initial prototype to build off of should be complete as well as all the Planning for the project through the canvas pages.
- Milestone 2: Each member completes first feature.
  - Date: 3/31/24
  - Goal: Each member completes one of there two features.
- Deadline 1: Testing, Coding, Documentation of first feature is complete.
  - Date: 4/7/24
- Milestone 3: Each member completes second feature.
  - Date: 4/14/24
  - Goal: Each member completes the coding of the second of there two features.
- Milestone 4: Testing, Coding, Documentation of second feature is complete.
  - Date: 4/21/24
  - Goal: All tests, code and documentation of the second feature should be done by this point.
- Deadline 2: Entire Project Should be finished.
  - Date: 4/28/24

Risk Analysis

(3/24) - unavailable

(3/29 - 3/31) - unavailable

1. Week of 2/26 - Midterm Exams
2. Classes and Work Risk Analysis
   M - work/classes (6-3)
   T - classes (12-4)

W - work/classes (6-11)
Tu - classes (12-4)
F - work (5-1am)
Sa - unavailable after 5pm
Su - work (7am-3)

Note: I am always available to respond to messages/emails

Project Progress

Week 1

Summary

- Completed Canvas and GitHub setup
  - Forked GitHub branch
  - Added feature plans
  - Added milestones/deadlines

Progress

- LoC: N/A

Week 2

Summary

Risk analysis - no work completed

Week 3

Summary

- Completed Subtasks feature

Milestones

- Milestone 2 complete

Progress

- LoC: 400

Week 4

Summary

- Wrote tests for subtask feature

Milestones

- Deadline 1

Progress

- LoC: 283

Week 5

Summary

- Finished feature 2
- Wrote tests for feature
- Finalized design and user manual for features

Milestones

- Milestone 3
- Milestone 4

Progress

- LoC: 700