San Francisco State University
SW Engineering CSC648/848
Milestone 4
Section 04 Team 06 [Check With Manuel]
StudyBuddy
05/08/2024

Team Members

Ashley Ching	Team Leader
Yuquan Xu	Front End Lead
Kent Nguyen	Back End Lead
Nhan Nguyen	Git Master/Front End
Brenden Lapuz	Scrum Master
Pierre Harbin	Team Member/Back End

1. QA Testing

1.1. Unit Test (Select 5 P1 Features to be tested):

- Login (backend) - The login component manages user authentication by validating input,
  - The unit test verifies the functionality of the backend handleLogin function.
  - It ensures correct behavior under various scenarios:
    - Valid login attempts.
    - Invalid login attempts (incorrect password or non-existent user)
    - Error handling cases
  - Functional Coverage:
    - Estimated to be around 90% due to comprehensive verification of various l
    - Ensures reliability of the authentication process.
  - Statement Coverage:
    - Estimated to be around 90%.
    - Tests exercise each line of the handleLogin function.
    - Demonstrates thorough code path coverage.
- Registration (backend) - The registration feature ensures smooth user sign-up, validating
  - The unit tests cover these functionalities by:
    - Verifying successful registration of a new user.
    - Testing proper handling of existing user emails.
    - Ensuring the creation of guest users.
  - Functional Coverage:
    - Validates user registration and guest user creation, including scenarios like
    - Coverage is estimated to be over 90%.
  - Statement Coverage: Ensures extensive testing of code paths within handleRegis
    - Includes different branches of conditional statements and error-handling co
    - Coverage estimated to be over 80%.
- Login (frontend) - Login feature in a React application, enabling users to input their crede redirects to a dashboard page upon successful login.
  - Checking if the Login component can render without throwing an exception
  - Mocked Dependencies:
    - Mocks the useNavigate and useAuthContext hooks to simulate their behav
  - Specific Test:
    - Renders the Login component using the render function from @testing-libra
  - Functional Coverage:
    - High coverage, around 95-100%.
    - Verifies the fundamental functionality of rendering the Login component wit
  - Statement Coverage:
    - Estimated to be moderate, around 50-70%

        - Primarily focuses on rendering the component, not targeting specific code p
- Registration (frontend): - allows users to input their information for account registration in
    - Verifies that the Registration component can render without throwing exceptions
    - Mocks dependencies such as `useNavigate` and `useAuthContext` to simulate the
    - Functional Coverage
        - High coverage, approximately 95-100%
        - Validates the fundamental functionality of rendering the Registration compo
    - Statement Coverage
    - Moderate coverage, estimated around 50-70%
    - Focuses on rendering the component, without targeting specific code paths.

1.2. Integration Test (For at least 10 P1 features, perform the below):

Integration Test: CSC 648 Section 4 Team6 Integration Testing

Most features tested were working to a satisfactory level. We have some more features that needed a lot more work so it was not included in the integration test. Each integration test is in a separate tab on the google sheets.

2. Coding Practices

2.1. A coding style

- We're currently following the Google JS coding style, enforced by the Visual Studio Code Prettier plugin. At the moment our file naming conventions do not follow this style, but we will change that shortly after this milestone
- Please list source files related to 5 P1 features which demonstrate your coding style.
    1. Login.js
    2. Registration.js
    3. MyGroups.js
    4. MyFriends.js
    5. Profile.js