Assignment 6

Hognogi Ana-Maria Cristina

Testing is an essential component of software development, employed to ensure that applications perform as expected under various conditions. It is applied at different stages of the development cycle to identify and resolve defects early, enhance product quality, and validate functionality, performance, and security requirements. Testing is useful because it helps prevent future failures, reduces maintenance costs, and increases user satisfaction by delivering a reliable and functional product.

In my bachelor's thesis, the web application developed facilitates the adoption of cats, aiming to streamline the process through a user-friendly interface and robust backend management. Features of the application include a searchable database of cats available for adoption, detailed profiles for each cat, user registration and login functionality, and the ability to request more information. The application is built using Node.js with Express for the server-side logic, Sequelize as the ORM for database interactions, and SQLite for data storage, while the frontend is developed with React and managed through Vite for efficient bundling and development experience.

**Unit Testing**

For unit testing the application, Jest is an ideal choice. It is a popular JavaScript testing framework that works well with Node.js and React. Its features support both synchronous and asynchronous tests, which are critical for validating functions involving database operations and frontend component behaviors.

* **Backend (Node.js with Express and Sequelize):** Test individual functions such as user registration by mocking Sequelize models to simulate database interactions. Use Jest to assert that functions behave as expected when given valid and invalid inputs.
* **Frontend (React with Vite):** Use Jest in combination with React Testing Library to test individual components in isolation. For instance, ensure that the user registration form renders correctly, handles state updates when users input their data, and triggers the correct function upon form submission.

**Integration Testing**

Integration testing can be managed using a combination of Jest and Supertest. Supertest is useful for testing HTTP APIs and can be integrated seamlessly with Jest to handle tests that require interaction between different modules like the backend server routes and middleware.

* Test interactions between user authentication routes and database access layers.
* Validate the integration of user session management with profile viewing and editing features, ensuring that security constraints are respected.

**System Testing**

For comprehensive system testing, Cypress is an excellent tool because it allows for end-to-end testing in real browser environments. It is effective for testing the full flow of the application from frontend interactions through to backend API responses and database integration.

* **End-to-End Workflow Testing:** Test scenarios such as user registration, user login, searching for cats, viewing cat details, and the adoption process.
* **Performance and Load Testing:** While Cypress primarily focuses on functional testing, tools like Artillery or k6 can be integrated to simulate high traffic and interactions to test how the system behaves under load. These tests should ensure that the application can handle multiple simultaneous user interactions without significant performance degradation.