

Webapp Developer Intern at ERIC Robotics, Premier Seals India Pvt. Ltd.

Internship Assignment:

Duration: 3 days

Problem Statement #1:

Your task as a Webapp Developer Intern is to design and develop a WebSocket API with RESTful endpoints and implement security features to handle a stream of string data received from a React application. The string data represents user input received from a simple joystick component in the React application.

Requirements:

1. Design a WebSocket API:

- Define the WebSocket API's architecture, including the necessary endpoints, message formats, and communication protocols.
- Decide on the technologies and libraries to be used for implementing the WebSocket API (e.g., Python3/Flask, Node.js, Express.js, Socket.IO)

2. Implement WebSocket API endpoints:

- Create WebSocket API endpoints that can receive and process the string data stream from a React application.
- Handle the connections, disconnections, and message transmission between the WebSocket server and the React application.

3. Develop RESTful endpoints:

- Design and implement RESTful endpoints to support additional functionality, such as user authentication, configuration settings, or data retrieval.
- Define the required API endpoints and their corresponding HTTP methods (e.g., GET, POST, PUT, DELETE).
- Ensure that standard error handling and response status codes for different scenarios are implemented.
- In essence, the user request will ping the RESTful endpoint, which then upgrades the connection to a full WebSocket stream.

Optional:

4. Secure the WebSocket API:

- Implement security measures to ensure, that only authorized users can access the WebSocket API and send/receive data.
- Choose appropriate security mechanisms such as authentication, encryption, and access control.

Problem Statement #2:

Now that you've developed the necessary backend software, you can start working on the client-side stack that will enable a user to interact with your joystick API.

1. Integration with React application:

- Develop a simple React application that includes a joystick component to capture user input.
- Connect the React application to the WebSocket API and establish a bi-directional communication channel to send the joystick data stream.
- Display received data or appropriate feedback from the WebSocket API on the React application interface or separately via a terminal.

2. Documentation and Presentation:

- Document the API design, including the WebSocket API architecture, RESTful endpoints, and security features.
- Provide a README file describing the instructions for running the WebSocket API server, React application, and any necessary setup or configuration steps.
- Present anything and everything you've developed during the assignment period in a GitHub repository.

Final Deliverables:

- Source code for the joystick WebSocket API server implementation, including the necessary configuration files.
- Source code for the React application with the joystick component.
- Documentation explaining the design and functionality of the WebSocket API and RESTful endpoints.
- Instructions for running the WebSocket API server and React application, including any dependencies or setup requirements.

Note: Feel free to use any additional libraries or frameworks as necessary to accomplish the task. Feel free to ask any clarifying questions before or during the assignment. Good luck with your assignment, and we look forward to reviewing your work!