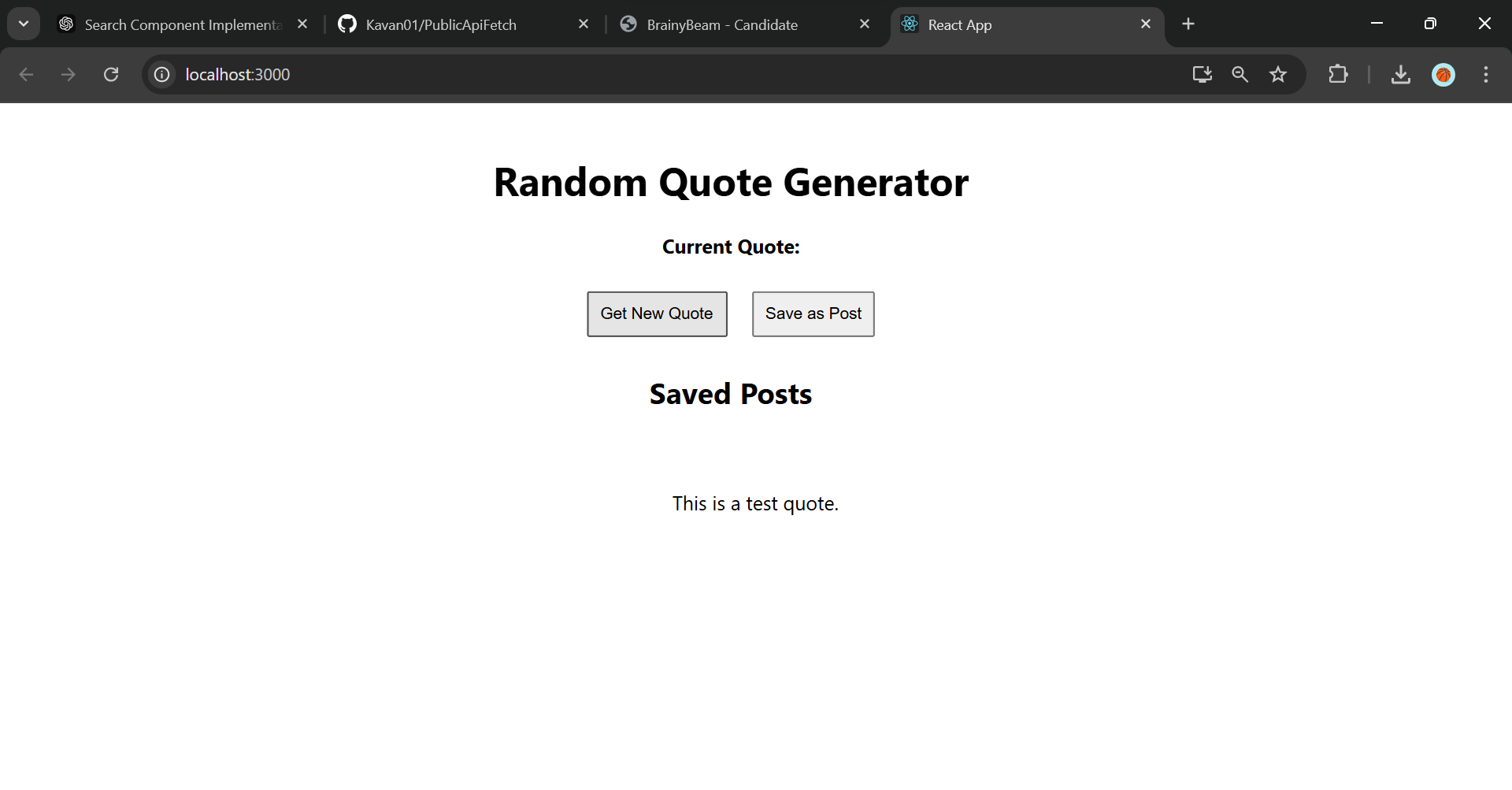
**Task Description**

Develop a full-stack quotes application using the MERN (MongoDB, Express.js, React.js, Node.js) stack. The application includes a server-side API to fetch quotes and a client-side React application to display and manage quotes. The following tasks were implemented:

1. **Server-Side Tasks**:
   * Set up Node.js and Express.js server.
   * Connected to MongoDB for database storage.
   * Created API endpoints for fetching, adding, and managing quotes.
2. **Client-Side Tasks**:
   * Built React components for displaying quotes and user interface.
   * Implemented Axios for API calls.
   * Handled CORS and server-client communication.
3. **Error Handling and Debugging**:
   * Resolved CORS issues by using a proxy server.
   * Debugged network issues to ensure seamless API communication.

**Task Output Screenshot**



**Widget/Algorithm Used in Task**

**1. Frontend (React):**

* **React Components**: Modularized components for header, quote display, and buttons.
* **Axios**: Used for making API requests to the backend server.
* **Bootstrap/Material-UI**: Styled buttons and layout for the user interface.
* **useState, useEffect Hooks**: Managed state and lifecycle events for fetching and updating quotes.

**2. Backend (Node.js, Express):**

* **Express.js**: Defined API routes for fetching and adding quotes.
* **Mongoose**: Connected to MongoDB and managed database schema for storing quotes.
* **CORS Middleware**: Enabled cross-origin requests between frontend and backend.

**3. Database (MongoDB):**

* **Quote Schema**: Defined the structure of the quotes stored in the database.
* **CRUD Operations**: Implemented create, read, update, and delete operations for quotes.