**Lifestyle Calendar Tracker**

**Concept**

Background: Lifestyle Directors and event coordinators have the responsibility of organizing events and special activities for their clients. This necessitates the ability to provide real-time updates and event forecasts. The aim of this project is to develop a web application that simplifies this process, offering a user-friendly interface and generating calendars to aid event tracking.

**MVP Feature Requirements**

As an anonymous user, I would like to:

* Create an account for event/admin user only
* Log in to an already-created account
* Add, change, delete events
* Track events and provide details of each event
* Provide a view of the calendar to guest viewers
* Contact event/admin user

**MVP Technical Requirements**

Based on your feature requirements, list out the major technical tasks that will need to be Tech stack:

* **Back end:**
  + **Server:** Node.js with Express.js
  + **Database:** PostgreSQL

**Front end:**

**Framework:** React

**Hosting and deployment:**

Further research needed for optimal hosting and deployment options.

**Backend:**

* Implement user authentication and authorization for event/admin users only.
* Set up RESTful API endpoints for CRUD operations on events.
* Utilize PostgreSQL database for data storage and retrieval.
* Ensure data security and integrity through proper validation and error handling.
* Implement logging mechanisms for tracking application activity and errors.

**Frontend:**

* Develop a responsive and intuitive user interface using React.
* Integrate with backend API endpoints to fetch and display event data.
* Implement user authentication and session management.
* Provide interactive event management features such as adding, editing, and deleting events.
* Design and implement a calendar view for displaying upcoming events.

**Rough Implementation Plan**

* **Week 1-2: Planning and Setup**
  + *Project Planning*
    - Define project scope, objectives, and requirements.
    - Create user stories and MVP feature list.
    - Determine technology stack and development tools.
  + *Environment Setup*
    - Set up development environment for backend and frontend.
    - Install necessary dependencies and libraries.
    - Initialize version control with Git and set up repository.
  + *Database Setup and User Authentication*
    - Set up PostgreSQL database.
    - Design database schema for users and events.
    - Implement user authentication and authorization using Node.js and Express.
* **Week 3-4: Backend Development**
  + *RESTful API Development*
    - Design and implement RESTful API endpoints for CRUD operations on events.
    - Implement validation and error handling for API requests.
  + *Event Management and Data Integration*
    - Develop functionality to add, edit, and delete events.
    - Integrate backend with frontend for event data retrieval and display.
    - Test backend functionality using unit tests and integration tests.
* **Week 4-5: Frontend Development**
  + *User Interface Design*
    - Design user interface mockups and wireframes.
    - Implement basic layout and navigation using React components.
  + *Event Management UI and Calendar View*
    - Develop event management features such as add, edit, and delete events.
    - Implement calendar view to display upcoming events.
    - Test frontend components and UI interactions.
* **Week 6-7: Integration, Testing, and Deployment**
* *Integration and Testing*
  + Integrate backend and frontend components.
  + Conduct end-to-end testing to ensure seamless functionality.
  + Address any bugs or issues identified during testing.
* *Deployment and Finalization*
  + Deploy application to chosen hosting platform.
  + Perform final checks and optimizations for performance and security.
  + Document installation, setup, and usage instructions.
  + Conduct user acceptance testing with stakeholders.
  + Prepare for launch and announce availability to users.