Project Report: Event Recommender Web App

# 1. Project Overview

This project is a location-based event recommender web application that enables users to discover events happening in various cities. The application includes a React.js and TailwindCSS frontend that communicates with a Flask backend. Event data is dynamically fetched via scraping from Eventbrite, ensuring updated and relevant content.

# 2. Technology Stack

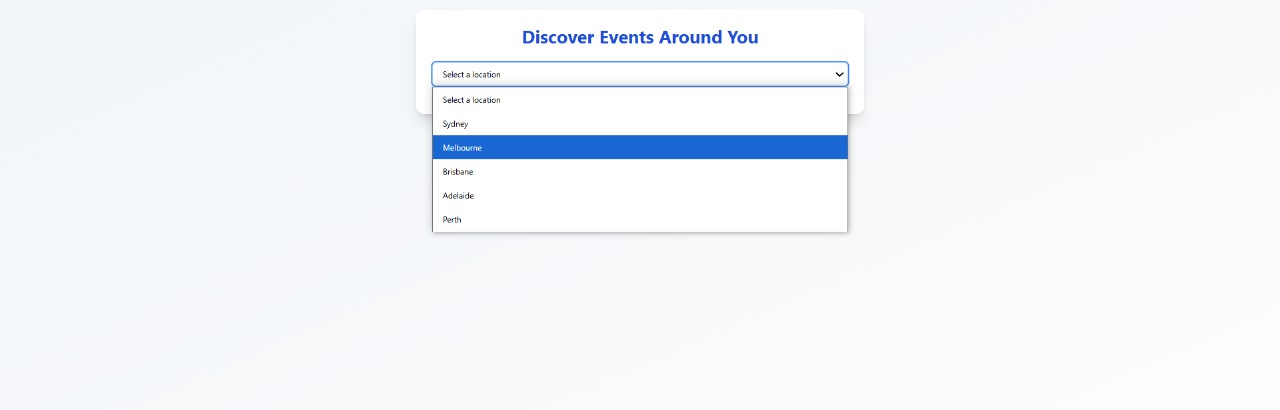
- Frontend: React.js, TailwindCSS

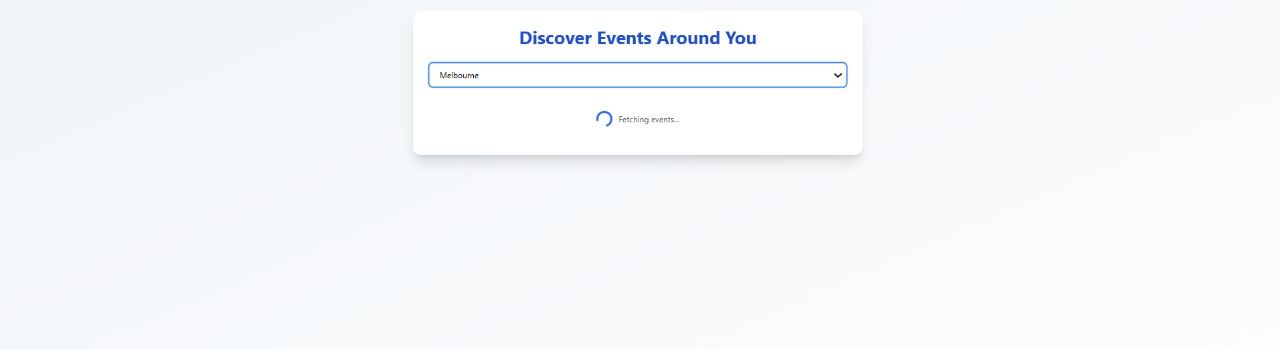
- Backend: Flask (Python)

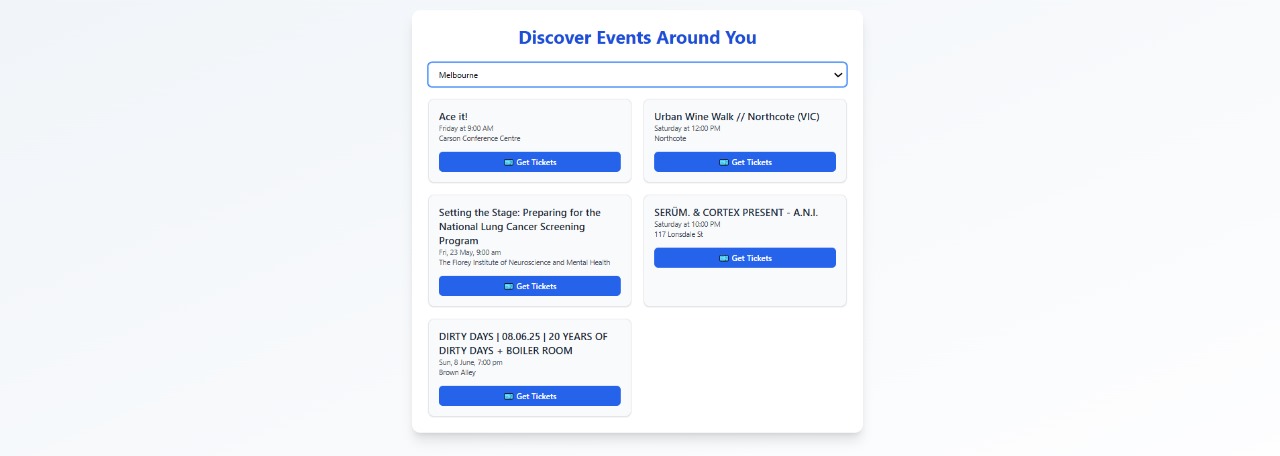
- Data Collection: BeautifulSoup, Selenium

- Deployment: Localhost (MVP phase)

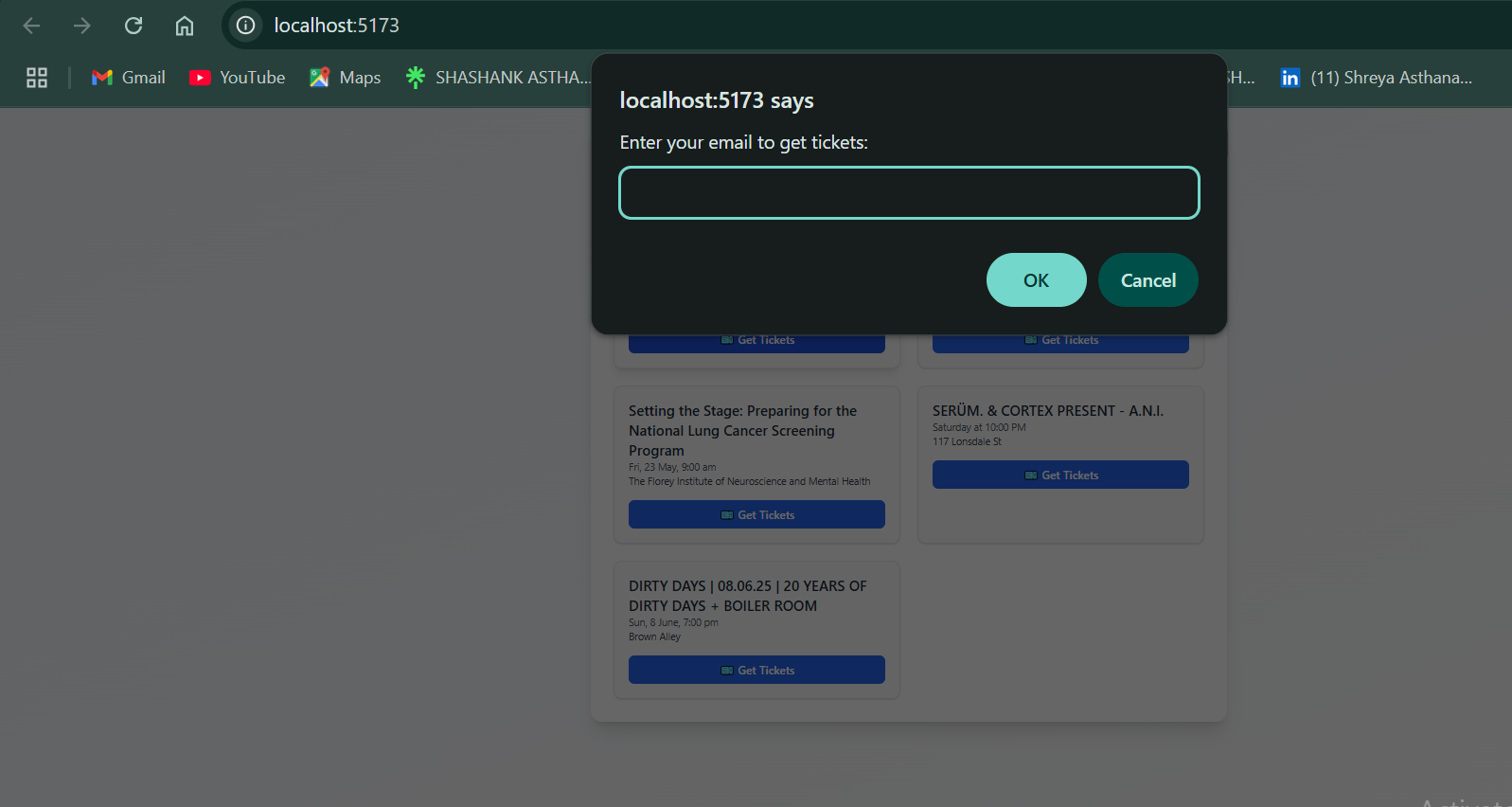
# 3. Key Features

- City-based event filtering with dynamic dropdown  


- Real-time loading animation during data fetch  


- Event listing with title, date, location, and ticket link  


- Responsive UI using TailwindCSS

- Email prompt before redirecting to event ticketing page  


# 4. Development Approach

1. Data Collection:  
 - Scraped event data from Eventbrite based on selected cities.  
2. Backend API:  
 - Flask served two endpoints:  
 • /api/locations: Returns available cities.  
 • /api/events?city=: Returns events for the selected city.  
3. Frontend:  
 - React with useEffect and useState for fetching and managing state.  
 - TailwindCSS provided styling and responsiveness.  
 - Conditional rendering used to show loading animation or events.

# 5. Challenges Faced

- JavaScript-rendered pages required Selenium over BeautifulSoup.

- CORS issues between frontend and backend resolved using Flask-CORS.

- Ensuring up-to-date data due to lack of persistent storage.

- Simplistic initial UI due to MVP focus and time constraints.

# 6. Future Improvements

- Add user authentication and personalized dashboards.

- Implement a database (e.g., MongoDB) for storing events and user preferences.

- Automate data scraping using scheduled tasks or Celery.

- Integrate LangChain and Twilio to send WhatsApp notifications based on user preferences.

- Improve frontend design and add filtering/sorting features.

# 7. Conclusion

The MVP demonstrates a practical implementation of a city-specific event discovery platform with modern web technologies. It serves as a solid foundation for building a more intelligent, AI-powered assistant that notifies users about relevant events via messaging platforms.