Headstarter Web Application Project Document

Project Overview

Project Name: Headstarter Web Application

Track: Track A

Team Members:

- **Shivang Raikar:** University of Massachusetts Boston
- **Ghady Abboud:** Drexel University
- **Duaa T:** CUNY, The City University of New York
- **Shubh Shahra:** University of Toledo

Fellowship Goals:

- Achieve 1000 people on the waitlist, 1000 accounts created, or $1000 in revenue generated.

Unique Selling Proposition (USP)

1. **Team Profiles and Project Showcases:**
   - Detailed team profiles with bios and skill sets for each member.
   - Each project will have a dropdown containing technologies used, a link to GitHub, and a link to the website.
2. **Gamification and Rewards:**
   - Gamification elements such as badges, leaderboards, and rewards for milestones achieved (e.g., most rated project, fastest growing team).

Project Features

1. **Team and Project Information:**
   - All teams with their respective tracks chosen.
   - GitHub and LinkedIn links for each user, and team links to their project GitHub and project website.
   - Rating system where each user can rate every project, with an average rating displayed beside each project.

Technology Stack

Frontend:

- **ReactJS**
- **HTML5 and CSS3**
- **Bootstrap or Tailwind CSS**
- **Axios or Fetch API**

Backend:

- **Node.js with Express.js** or alternatives (Django, Flask, Spring Boot)

Database:

- **MongoDB** or **PostgreSQL**
- **Firebase** for real-time data handling and authentication

Authentication:

- **JWT (JSON Web Tokens)**
- **OAuth2** for third-party authentication providers

Hosting and Deployment:

- **AWS** (EC2, S3, RDS)
- **Heroku**, **Netlify**, or **Vercel**

Version Control and Collaboration:

- **Git**
- **GitHub**

Additional Tools:

- **Webpack** or **Parcel**
- **Redux** or **Context API**
- **Socket.io** (if real-time features are needed)

Project Steps and Milestones

1. **Week 1-2:** Finalize project requirements and design.
2. **Week 3-5:** Develop frontend and backend components.

3. **Week 6-7:** Integrate database and authentication.
4. **Week 8:** Testing and bug fixing.
5. **Week 9-10:** Deployment and user feedback collection.

Chatbot Implementation with RAG

Objective: Create a chatbot using Retrieval-Augmented Generation (RAG) to assist with queries related to the project setup.

Implementation Steps:

1. **Define Chatbot Scope:**
    - Handle queries related to project status, team member details, and technical questions.
2. **Build the Retrieval System:**
    - **Data Collection:** Gather relevant data and documents.
    - **Indexing:** Index data for efficient retrieval.
3. **Integrate RAG Model:**
    - **Retrieval Component:** Implement a system to fetch relevant documents.
    - **Generation Component:** Use a language model (e.g., GPT-4) for context-aware responses.
4. **Develop Chat Interface:**
    - **Frontend:** Build a chat interface integrated into the web application.
    - **Backend:** Connect the interface to the RAG model and retrieval system.
5. **Testing and Iteration:**
    - **Test Functionality:** Ensure accurate query handling.
    - **Collect Feedback:** Refine based on user feedback.
6. **Deployment and Monitoring:**
    - **Deploy:** Integrate or launch as a standalone service.
    - **Monitor Performance:** Track and adjust as needed.

Meeting and Action Plan

Meeting Platform: [Specify Platform]

Agenda:

1. Introductions
2. Fellowship Goals and Strategies
3. Roles and Responsibilities
4. Project Plan
5. Communication and Check-ins
6. Action Plan Submission

Communication and Check-ins:

- Primary communication via Slack.
- Weekly virtual check-ins every Saturday, 7 pm or random days whenever required.

Action Plan Submission:

- Submit a concise action plan summarizing goals, strategies, roles, milestones, and communication plans.