

ENSF608 Project Proposal

Group 1

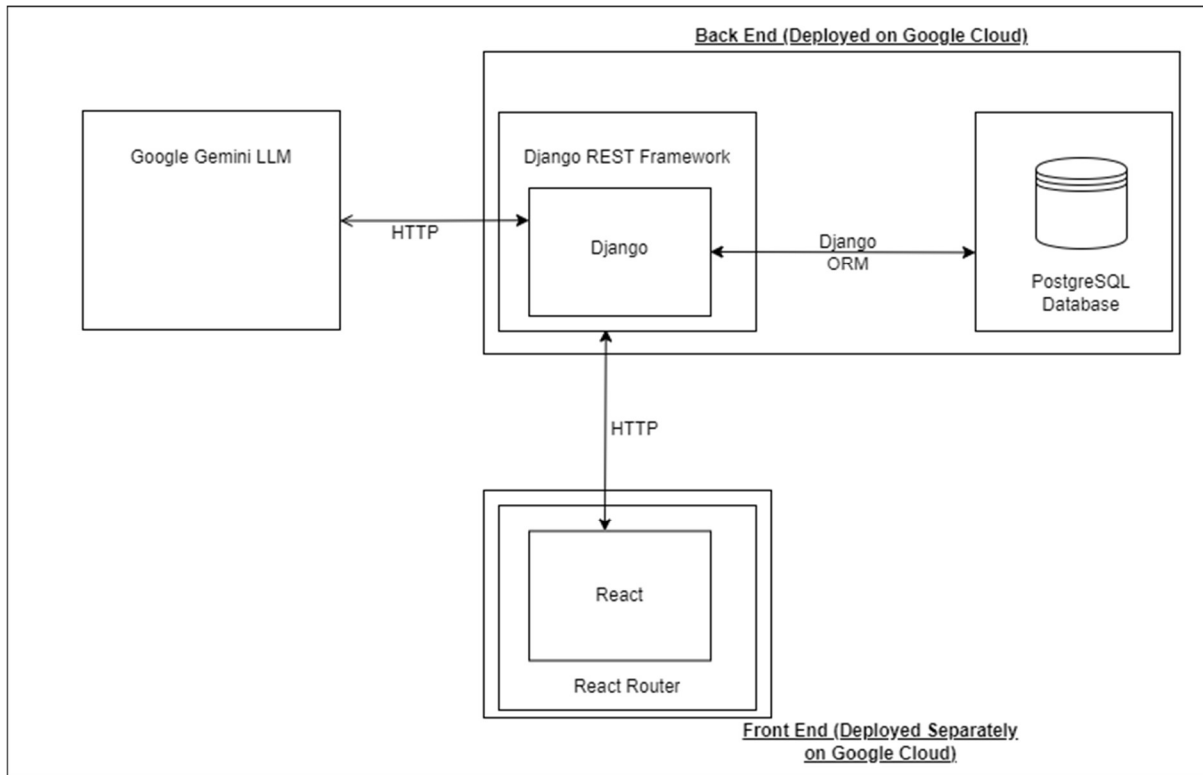
Our web-based application, Large Lifting Model (LLM App), aims to help gym goers by creating custom workout plans tailored to the users abilities and personal goals. This app will be useful for a wide audience: from novice lifters who don't have the knowledge to create a plan, to experienced athletes who don't have enough time or money to get custom workouts. The custom workouts are driven by a large language model (LLM) and uses user preferences that are setup via the web interface to guide the workout customization.

The core feature of our application is its ability to provide users with custom tailored workout plans, specifically designed to meet their individual fitness goals, physical abilities, and preferences regardless of their level of experience. This is enabled in two ways, through a highly customizable user preference and workout preference screens which allow for a selection of many different parameters (workout style, duration, equipment, etc.), and by a rolling workout history to track what the user has done in the past. The front end will be responsible for the preference selection and will be implemented using React and the React router framework. We've chosen React as it is a powerful JavaScript library that is designed to create user interfaces for web applications and we have some experience with it. Our application will feature several different pages (login, user profile, workout generation, workout history, etc.) and will work with multiple unique users, all of which will be handled through the React library.

The React frontend will be supported by the Django backend framework, the Gemini LLM, and a PostgreSQL Database all hosted on google cloud. We have chosen to use Django and the Django REST framework as we have some experience with it and so that we can access our backend using an application program interface (API) format. The backend will be responsible for accessing the database, business rules, and interfacing with the LLM. The driving force of the project is the LLM which acts as the workout generator. We've chosen to use Google's Gemini for our LLM as it has an accessible API which will work well with the Django REST framework, and is very well documented which will help with a faster implementation. The LLM will have access to the user's previous workouts (through Django), enabling it to tailor newer workouts based on what the user has already done before. Finally, all the user personal and workout information will be stored in a PostgreSQL database. This is a powerful relational database that is commonly used with Django applications, reducing stability risks that could arise in our application. The LLMapp layout is shown in the diagram below.

Our team is divided into three categories: front end, back end, and project management. Each team has a team leader who is primarily responsible for the deliverables of their respective group and every member chooses a primary group to join. To promote portability, team flexibility, and equality of learning (i.e. being able to work with the project technologies) everybody also takes on a minor secondary role involving coding or project management within their team. For example, a back end developer would also be accountable for project documentation, and someone on the management team will take on back end or front end deliverables. Our project documentation is

stored on github (with merges QA'd and managed by team leads) and we are using JIRA for Kanban style project tracking.



LLMapp Diagram