Capstone 2: Project Proposal

EASYGPT

Created by:

Derek Russel

Developer(Intern)

Prepared for:

Timmy Glynn

Software Engineer

Executive Summary

Objective

Build a database-driven website off an external API or an internal API that the developer creates themselves. This project will incorporate all the skills learned over the bootcamp to create a full stack application.

Goals

- Build a full stack web application
- Select an app idea that will allow me to showcase full stack skills
- Select an app idea that will allow me to showcase full stack skills
- Define app features and functionality
- Use existing API

Purposed App Idea

I have always been fascinated with the AI technologies, as they develop and become more relevant in our everyday life the more interested I become about learning the capabilities and limitations of AI. One great way of learning more I thought would be to create an application that uses ChatGPT API. This application will make creating prompts to the Chat AI easier. Currently, you need to be very specific in your request to recieve a helpful response. My application will be easy to use form with selections that allow more vague questions to be responed to as if they were more specific questions.

Project Outline

Step One: Initial Project Ideas

You'll pick up to 3 project ideas to propose to your mentor and the community. You'll also explore some potential APIs.

Step Two: Project Proposal

For this step, you'll write a proposal for the site you want to build. This will help your mentor better understand your chosen capstone project idea.

Step Three: Source Your Data

After your mentor approves of your capstone project proposal, you'll figure out the database design of your application and where your data will come from. You may choose to use an existing API or create your own.

Step Four: Coding User Flows

Once you've figured out what you're building, you'll write the code to implement it. It's important to think about what you want a user's experience to be like as they navigate your site.

Step Five: Polishing Your Application

Once you have the core functionality implemented, you'll focus on additional UI enhancements and styling for your application. This is where you will implement your bells and whistles.

Step Six: Documenting and Submission

You've done a lot of work so now it's time to show your mentor your progress! Create a README in markdown, make sure your GitHub is organized, and submit your finalized project.

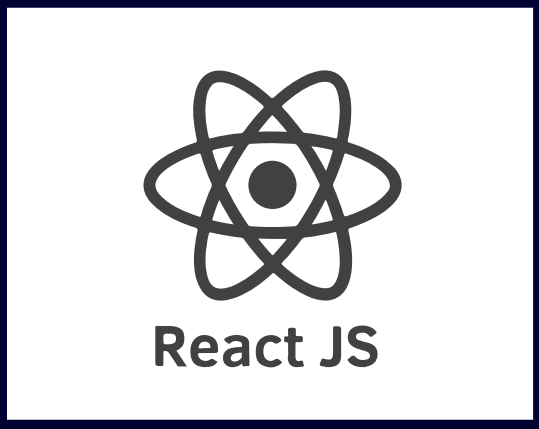
The Tech Stack

Back-End

For the back-end I will be using Node.js and Express. The reason for choosing this over Python is to keep my app in a single programming language. Also most fields I'm looking to apply my skills will be in web-development and in the Javascript language. I will be using the API from openAI where the chatGPT API is available. I will make requests to the API using axios library.

Front-End

For the front-end, which will be the main focus of this application, I will be using React.js. I choose react because of the way I will be able to create components, which I beleive will be beneficial when it comes to make the forms on the seperate pages. I will be re-using much of the same code so it seems beneficial to use components.



The App Details

What is the App?

The main use of the app will be to simplify the process in which to make responses that are valuable using the ChatGPT AI. For example someone might ask "Create a post about productivity." This isn't quite specific enough for the AI to give the best results. So I would send "Create a LinkedIn post under 200 words about the importance of productivity and time management for small business owners." To do this I would make a form they can use, with drop down options of what kind of information they are looking for.

App Creation

To being creating the app I will be using the npx create-react-app command from terminal. This will give me all the necessary files I need to being development of my app. From there I will use API calls using Axios and front-end routing will be done with React Router.

Data Use

I will be using the data from ChatGPT to populate the responses, I will also create a database to store user information such as username,password and history of requests and the responses received.

User Demographics

Student

- Age Range: 18-40
- Learning new skill that has questions.
- Needs to translate languages
- Essay formatting or cite formatting.

Developer

- Age Range: 18-60
- Help with algorithms or methods.
- Debugging issues with functions.
- Converting code to more efficient code.

Teacher

- Age Range: 20-70
- Creating seating charts.
- Generating word lists
- Creating fill in the blank assignments.

Content Creator

- Age Range: 20-70
- Generating ideas for content.
- Show casing AI technology.
- Creating funny stories to share.

Project Approach

Database Schema

I will create the following tables: Users, Requests, Responses. The Users table will contain user information such as userid, username, password. Requests will have a relationship with the User table to have a foriegn key matching the userid, this will allow me to show users which requests they have made. Requests table will contain the prompt to the API and also an ID. The Responses table will have a foriegn key from the Requestsid to match the responses with the prompts. This will also show under the user.

Possible Issues

There might be issues with the API requests and the limitations of the API account. This may include limitations on the type of response and also the permissions the API allows.

Sensitive Information

The most sensitive information will be the user's password. This will be encrypted when it is sent so it will not be seen by anyone trying to manipulate the site. No user sensitive information will be held like name, address, email.

App Functionality

The app will include components, forms, input fields, the ability to sign in and sign out. View history or requests and responses.

User Flow

User will come into the homepage and sign in, after sign in they will be brought to profile page where they can change password and view previous requests or responses. At the top will be a nav bar where they can make a new request which will bring them to a page with widget(icons) that they can choose an option. From there they will be brought to a form that has basic information and drop downs to choose from to create a prompt.
