

Frontend Generator

Screenshot to React JSX + CSS

Submitted By: Bhavuk Sagar (Bhavuksagar@gmail.com)

Problem Statement

Create a basic AI-powered tool that can take a screenshot of a webpage design and automatically generate the corresponding React JSX and CSS code.

Objective

Build a simple AI-powered Streamlit web app that:

- Accepts a user-uploaded screenshot of a webpage design.
- Sends the image to Gen-AI Model.
- Returns and displays the generated React JSX and CSS code.

Tech Stack

Followings are the tech and tool which are used in creating the app:

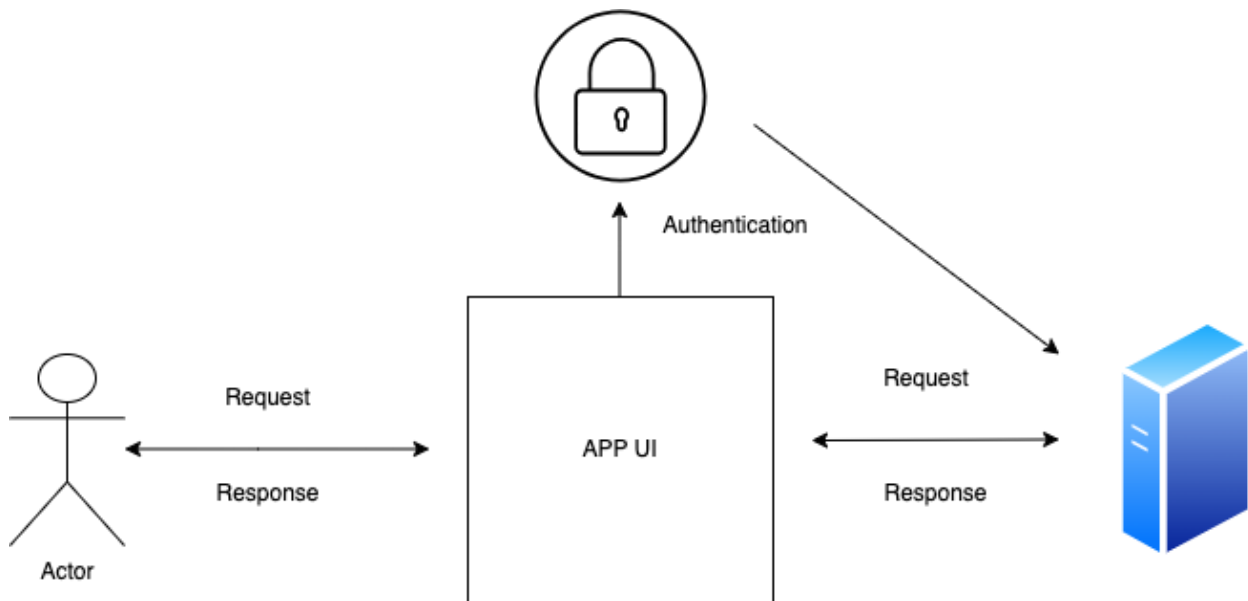
- **Frontend/UI:** Streamlit
 - **Backend/AI:** OpenAI GPT-4o (via API)
 - **Language:** Python 3x
 - **Libraries:**
 - `streamlit`
 - `openai`
 - `Os`
 - `Base64`
-

Setup instructions

1. Install all the libraries present in ***requirement.txt*** file.
2. Fetch the OpenAI API key and replace it in ***.env*** file.
3. Execute the following command in terminal to run the app

Streamlit run [*main.py*](#)

How It Works?



- UI takes the input from the user in the form of image and backend logic convert that image into base64 format suitable for API.
- OpenAI API key make the authentication with openAI server to process the request.
- API send the encoded image and the custom prompt to the GPT-4o model through the api and generate the code and send it back in the form of response to the app.

-
- Prompt used in code:

You are a frontend expert. Analyze the uploaded image of a web UI and generate the corresponding React JSX and CSS code.

Output only the code in clearly labeled code blocks and focus on layout/structure over exact pixel perfection.

- In model request we using the system role with custom prompt to assigning the role to the model which can't be override by the users prompt.

Limitations

Current version has following limitations:

1. **Single Input:** It takes only a single input image and makes the request.
2. **Billing:** Users need an OpenAI API key with GPT-4o access. Without sufficient quota or billing, the app will not function.
3. **Interaction with model:** Users cannot interact with model to make the changes in the code as it doesn't have a chatting interface.

Possible Improvements

1. **Interaction:** We can upgrade the UI and add chat space by which users can interact with model.
2. **UI Upgrade:** Add Reset button to clear the last response from the screen.
3. **Interactive Preview:** Render a live preview of the generated UI.