UI-Generator

Final Presentation

CSCI 2340 Spring ‘24

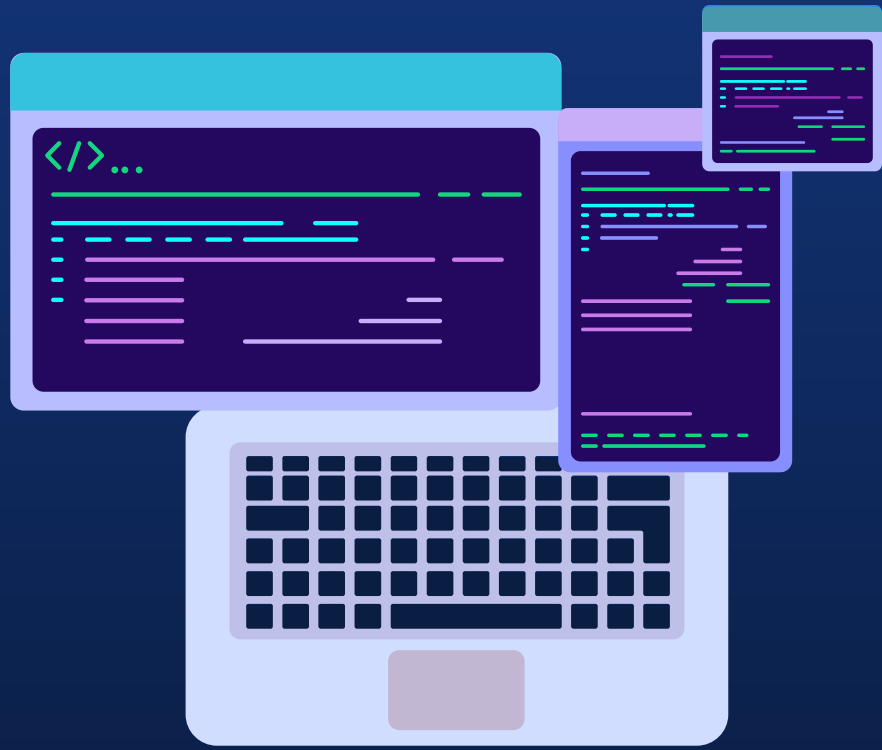


TABLE OF CONTENTS

01

Overall Status

Our Goal

A program that takes user input in a form and uses it to generate a web page and its source code that matches the user's description.

Overall Status

- Prior Deliverables ✔
- Design of Program ✔
- Most of the implementation ✔
- Enhancements (to be continued..)

02

Requirements

Requirements

- Generate a web page based on user input collected using a form
- Also generate HTML/CSS for the generated web page
- Allow iterative prompting to modify the results

Key Accomplishments

Database and API Integration

User auth, Postgres DB (code written, but not merged yet)
OpenAI API

Prompt Optimization

Developed and iterated on the prompts based on tests and feedbacks

Frontend Development

Developed the frontend for a form and a page displaying the generated web page/code and a chat window

Iterative Prompting

Created a package iteractive_prompting_client and used it in our software.

03

Specifications

Specifications

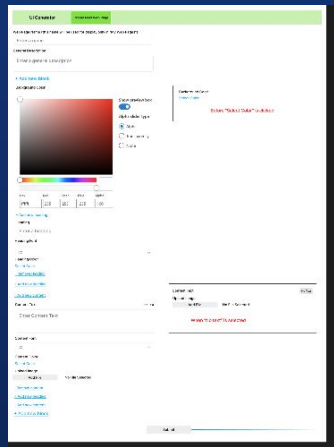
- Functional Specifications
  - *User Interface*: Provide a simple but friendly UI for the users to interact with our backend system
  - *Web Page Generation*: Provide the functionality to generate web pages and code based on user input
  - *Iterative Prompting*: Talk to openAI to iteratively refine the results
  - *Output Display*: Display the returned webpage/code to the users

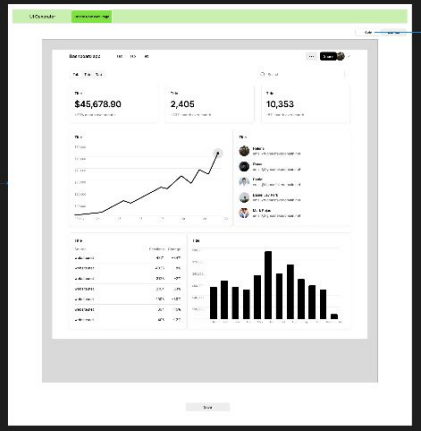
- Non Functional Specifications
  - *Performance*: Results should be generated relatively quickly

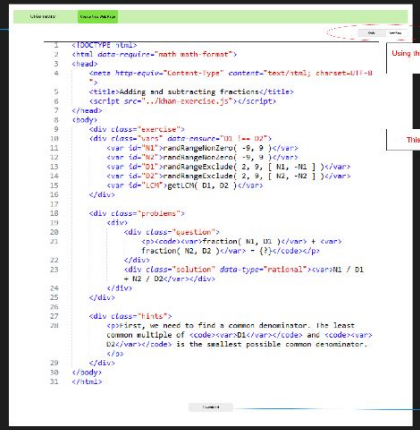
04

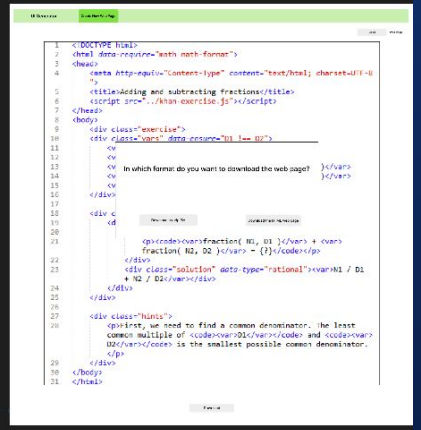
Design and Implementation

Initial UI design (we diverged from it in the implementation phase)



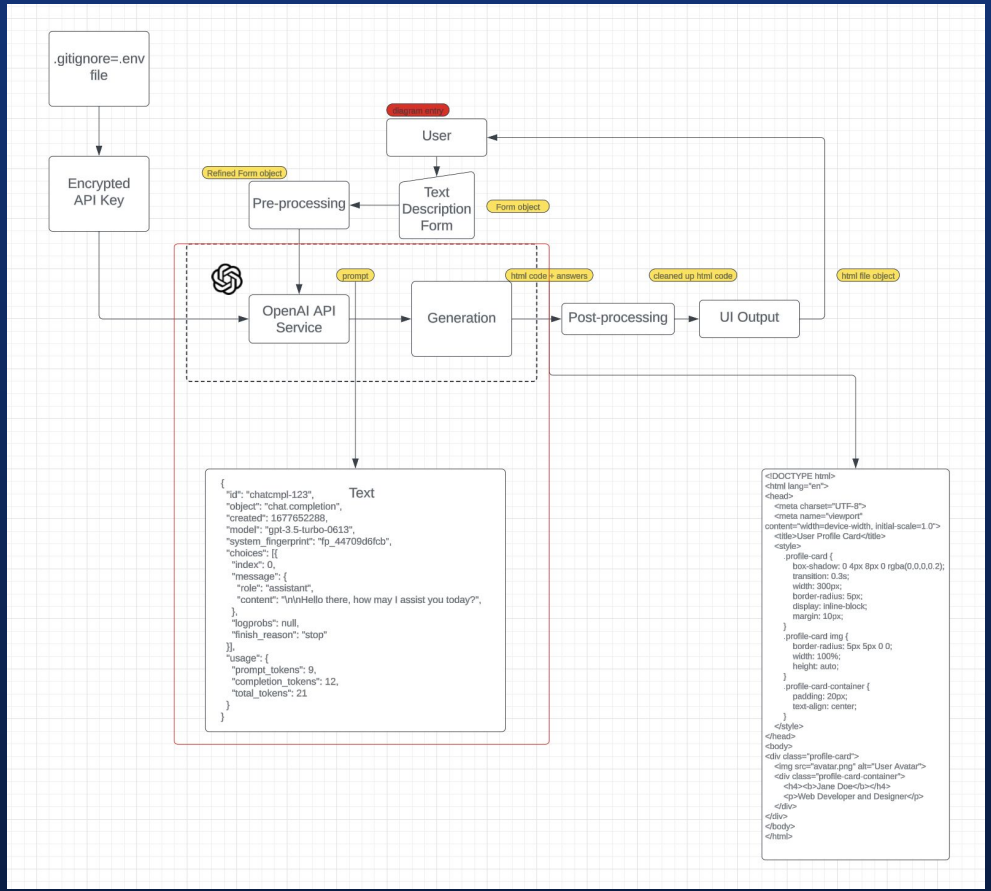






High-level system design

Iteratively prompt OpenAI API's GPT-3.5 turbo model with text input retrieved from the user until the image output (sketch) returned by GPT satisfies the user.



Implementation

- We built a next.js web application with frontend and prompt generation written in React and TypeScript
- We also published an NPM package for the API client, which communicates with GPT to generate results
- We implemented a form to collect user inputs and another page to display the generated web page and code, and a chat window to prompt GPT to refine the web page
- We also enhanced the UI with some animation

Demo

05

What we learned & What needs to be done

What we learned

- How to build a web application from end to end
- Many of us were not familiar with next.js and typescript but we learned along the way
- Prompt engineering is important to ensure that outputs are appropriate for the end-user
- Teamwork is important – Dividing up tasks helps improve efficiency, building on top of each other's work requires effective communication

What needs to be done

- Enhance form functionalities
  - Add more fields
- Enhance prompt quality
  - Experiment and find better ways to prompt GPT to get desired results
- A possible stretch goal is user authentication, which includes sign up/log in features and saving the generated UIs in the database
- Test the functionalities and robustness of the application
- Deployment

Thank You

CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon**, infographics & images by **Freepik**