itWorksOnMyMachine: Caden Khuu, Tiffany Yang, Wen Zhang, Qianjun Zhou
SoftDev
P04: Makers Makin' It, Act II -- The Seequel
2025-03-26
Time Spent: 3
TARGET SHIP DATE: 2025-04-20

DESIGN DOCUMENT (VERSION 1)

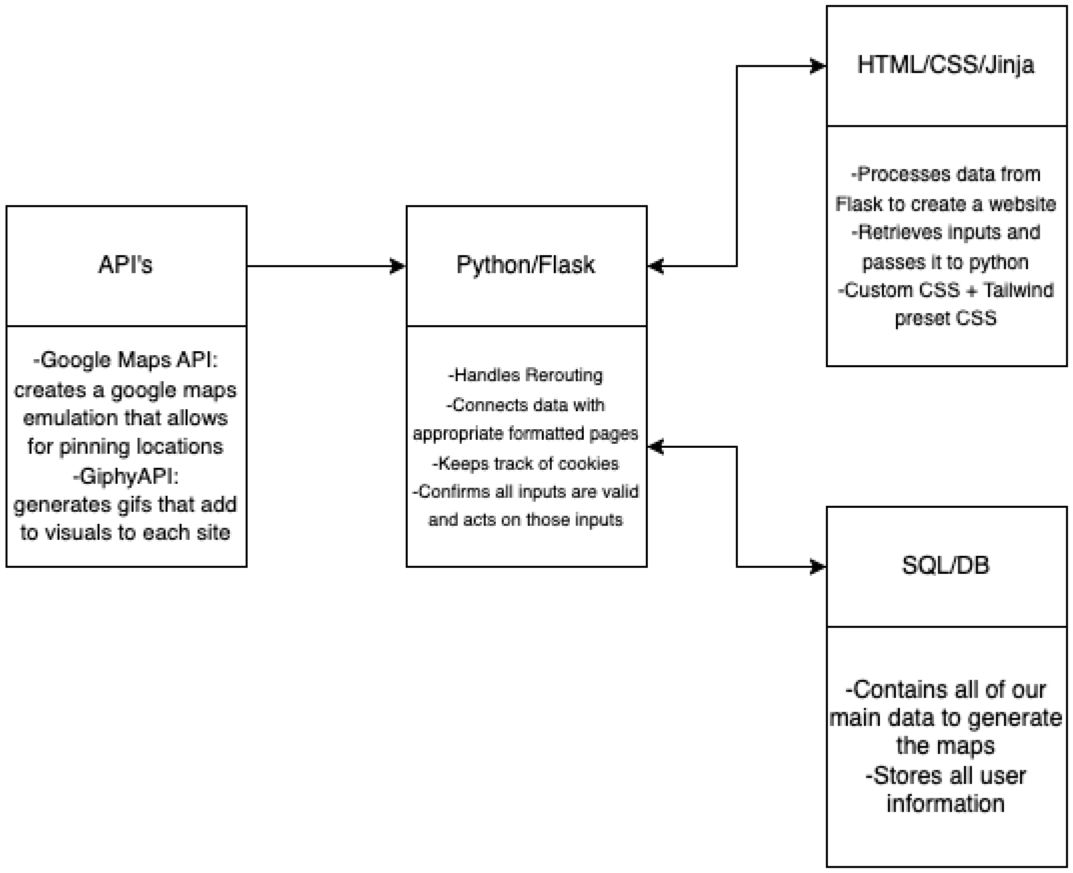
I. Description

This project is a recreation of google maps. There will be UNESCO heritage sites pinned on the map, clicking said pin will take you to a resource page detailing all that is needed to be known about the site. We will also include features that allow you to filter out certain sites and look for certain results in our dataset of heritage sites. When logged in, you will be able to favorite certain sites by saving it under your profile.

A. Program Components

a. User Accounts:
   i. Creation of accounts and login/logout functionality
   ii. Sessions to remain logged in for ease of use
b. Routes to different pages of the website using Flask and Python
c. API
   i. Leaflet API: Creation of the map similar to that of google
d. Data
   i. Stored in a csv file to be read in the python file.
   ii. Data can be found here: https://www.kaggle.com/datasets/joebeachcapital/unesco-world-heritage-sites/datad
e. SQLite3 Database: Stores data of the user and favorite sites
f. Jinja Templates:
   i. User dashboard:
      1. Logged-In State: Contains a profile image, a logout button, a setting page to change passwords and view favorites. As well as everything in the logged out state.
      2. Logged Out State: Contains only the map and being able to filter results on the page.
   ii. User Settings: Allows the user to edit their username, and password
   iii. Heritage Site Viewer: Displays the site as well as the corresponding data

B. Component Map

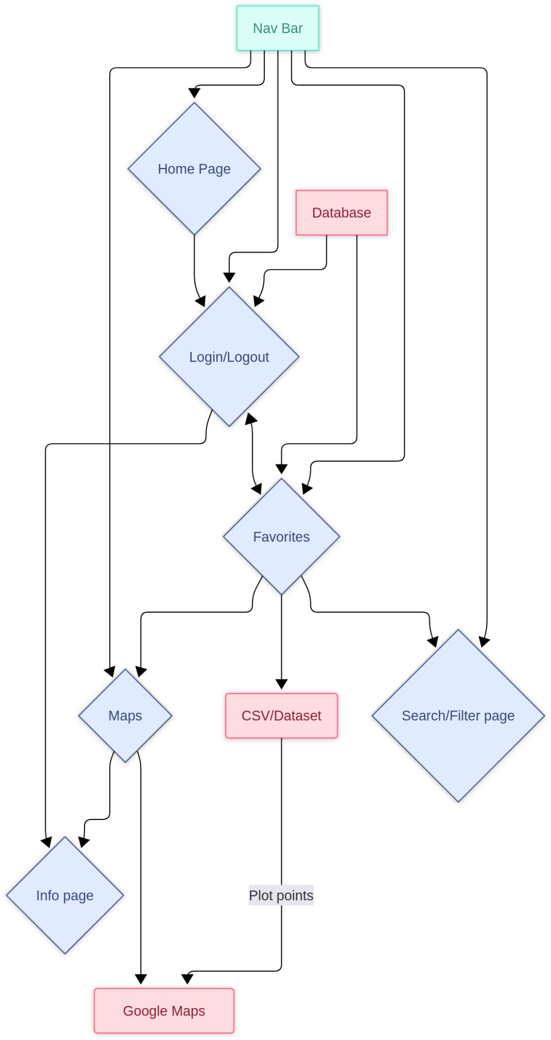

HTML/CSS/Jinja
-Processes data from Flask to create a website
-Retrieves inputs and passes it to python
-Custom CSS + Tailwind preset CSS
API's
-Google Maps API: creates a google maps emulation that allows for pinning locations
-GiphyAPI: generates gifs that add to visuals to each site
Python/Flask
-Handles Rerouting
-Connects data with appropriate formatted pages
-Keeps track of cookies
-Confirms all inputs are valid and acts on those inputs
SQL/DB
-Contains all of our main data to generate the maps
-Stores all user information

C. Program Component Connections

a. User accounts: Give access to favoriting sites and changing passwords.
b. Routes + Python: Routes allow users to traverse the website. They connect the different pages (HTML documents) of the website. Python also interacts with API and the database.
c. Database: Stores information related to the user (ID, Password, etc)
    i. One of the main factors for information exchange between components (has all the data)
d. APIs: Provide more information or other features that make the project more exciting
e. CSV file : Provides information to be displayed on the HTML page or be stored in the database

Templates: Allow for dynamic web pages

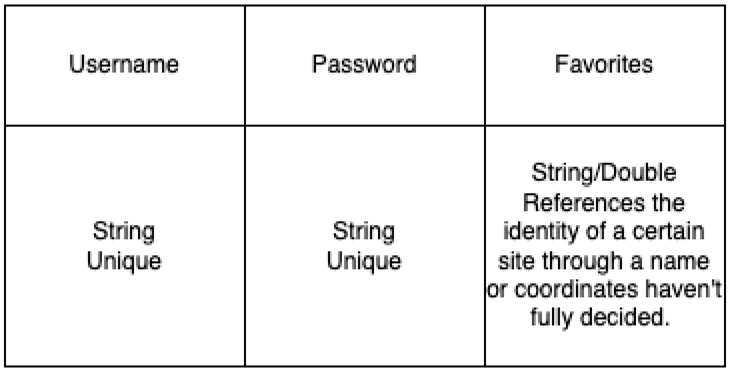
D. Site Map + Descriptions



A. Homepage: The first page that you are brought to.
B. Login/Logout: Allows you to log in and log out of your account when on this page.
C. Favorites: When logged in, this page will contain all the sites that you have favorited
D. Maps: The map that contains all the information and allows you to filter out certain sites through information on the sites to create a more personalized/useful map. Hovering over a location displays additional information on the heritage site. As well as any other information we have on that site. The popup also allows you to favorite it.

The Nav Bar allows you to have access to each of these sites.
The red boxes are all where the route is drawing the data to create it from.

E. Database Organization



Username	Password	Favorites
String Unique	String Unique	String/Double References the identity of a certain site through a name or coordinates haven't fully decided.

F. Program APIs

a. GiphyAPI: Generating related gifs for each site
    i. Probably won't but can probably generate gifs as part of the corresponding data for the heritage site.
b. Google Maps API
    i. Allows you to create the map of the world and set up pins where you want them

G. Frontend Framework: Foundation

1. Why tailwind?
    a. Easy-to-use and easy-to-follow tutorials
    b. Vast amount of CSS components and JS ones that will make things look and feel good
    c. Easy to change the list-style-type without writing extra css
2. How tailwind?
    a. It is very utility first which makes it easier to set margins, padding, colors, text-alignment, and others.
    b. Pre-built components like nav menu, forms, and cards.
    c. JIT mode which allows for only necessary css to be included

H. Data visualisation library

Why d3js?

-Highly flexible. We will be able to not only use preset charts but also any other visualization through directly manipulating certain elements.
-Supports working with our dataset (csv file).
-Allow for interactive features which might make the maps easier.
-Seems to work seamlessly with Flask and Python.

How d3js?

-Geographic maps that can help with understanding the google maps API
-Charts for favorited sites
-Selections and transitions for the filter

I. Task Breakdown

1. Caden Khuu (PM): Backend
    a. Create SQLite3 database schema
    b. Contribute to API interactions
2. Tiffany Yang: Frontend
    a. Create HTML templates with jinja templating: the forms required for logging in and signing up, the filter page to filter for certain results
    b. Design site style with CSS and Tailwind (our choice of front-end framework)
3. Qianjun (Ryan) Zhou: Backend
    a. Routing and logic between pages + User session management
    b. Linking database and API functions with site features
    c. Data processing to send over to HTML templates
4. Wen Zhang: Fullstack
    a. Starts off by helping backend setup whatever is necessary to get a working product
    b. Helps with whatever part is falling behind

QUICK RECAP:
Website that allows you to look up world heritage sites and information on those sites.
Framework: Tailwind
Frontend: HTML/CSS
Backend: Python/Flask
Database: SQLite
Visualization Library: d3js