Andrew Tsai, Mustafa Abdullah, Ricky Lin, Yu Lu

Snorelacks' Design Document Target Ship Date: 2026-12-22

SoftDev
P01: ArRESTed Development

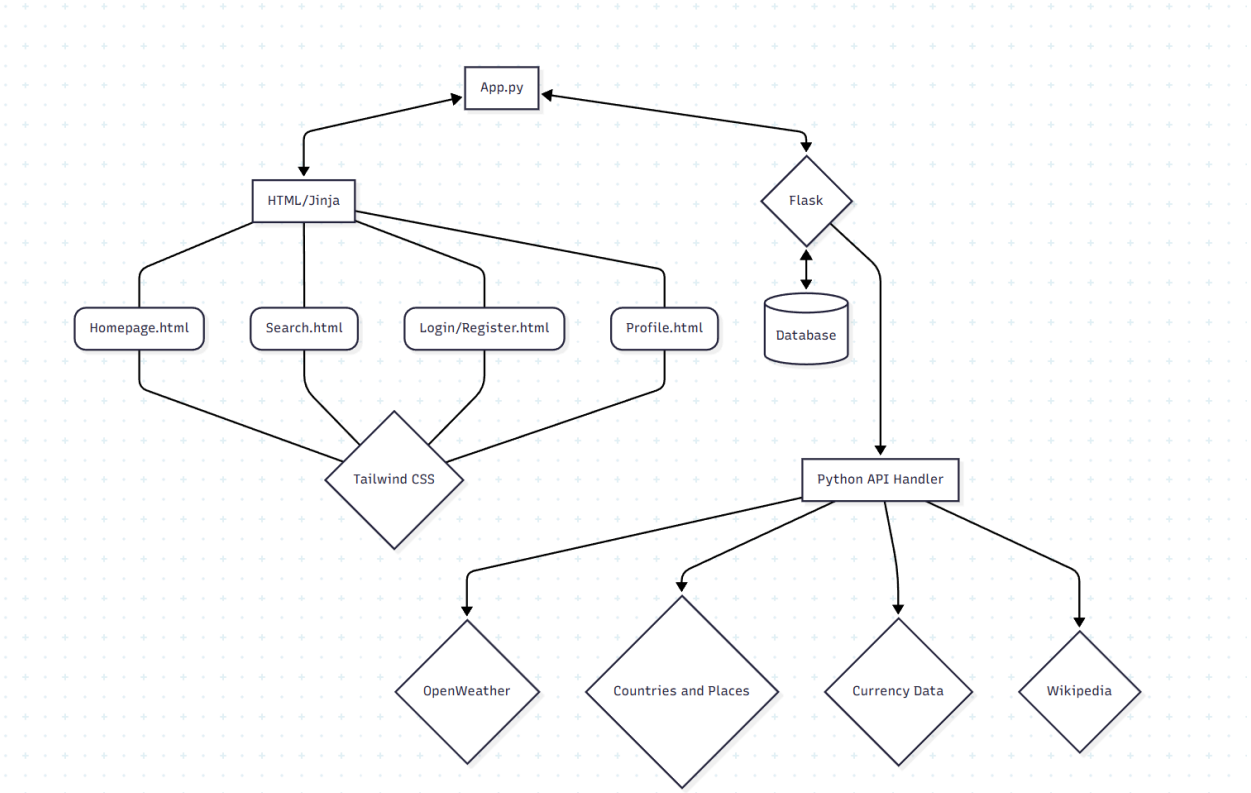
Travel Itinerary Website

Choose a country, see its flag, weather, find famous landmarks to visit, currency rates, population, possible routes to take; all to find your perfect vacation!

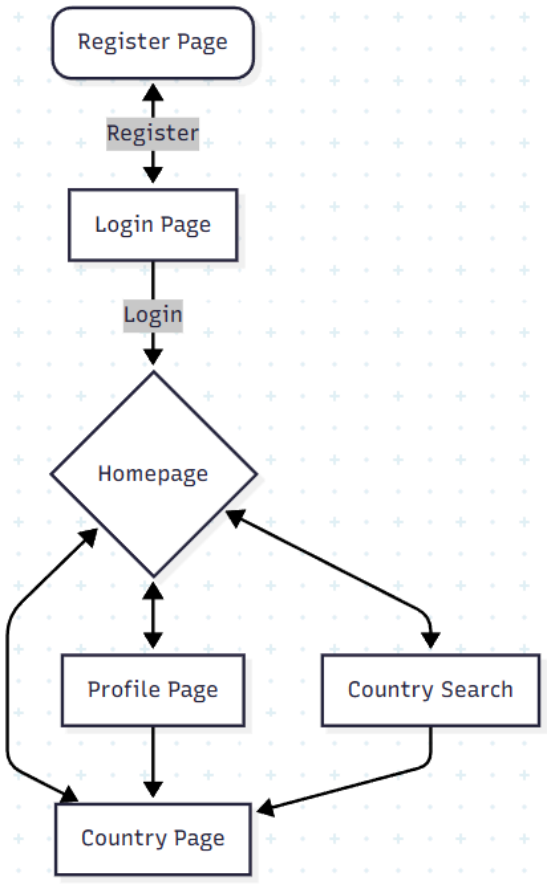
Program Components:

- Frontend
  - HTMl Templates
    - Country pages
    - Login Page
    - Register Page
    - Profile page
    - Homepage
  - CSS
    - *Tailwind* Frontend supported by *Jinja2*
- Backend
  - Database(Through SQLite3)
    - Python Backend handling updates and requests
    - User login DB
      - Stores Username, PW, Default settings (country of origin/address, default currency)
  - Flask Functionality (Through Jinja2)
    - Python Backend handling routing
  - API
    - OpenWeather
    - Countries
    - Currency Data
    - Wikipedia API
    - Places API

Component Map:



Sitemap:



Database Organization:

Users		
TEXT	Username	No duplicates, unique string
TEXT	Password	Password of the user
TEXT	Country of Origin/Address	The location of the user
TEXT	Default Currency	The currency that the user uses

Countries		
TEXT	country_name	No duplicates, unique string
TEXT	wiki_data	Json converted to string of wiki data
TEXT	country_data	Json converted to string of country data
TEXT	timestamp	Last updated time

Favorites		
TEXT	username	Username of the user
TEXT	country_name	country_name

API Section:

Countries API	OpenWeather	CurrencyData	Wikipedia API	Places API
https://restcountries.com/	https://openweathermap.org/api	https://marketplace.apilayer.com/currency_data-api	https://en.wikipedia.org/w/rest.php/v1/	https://apidocs.geoapify.com/docs/places/
Pull information about certain countries' population, capital, currency, etc.	Pull weather data based on current timestamp and defined location	Provides exchange rate for different currencies	Provides wikipedia pages based on queries	Provides places parsed by category
https://restcountries.com/v3.1/all Auth: None(Public) Parameters: name, all currency Endpoints: /all /name /currency /alpha	https://api.openweathermap.org/data/2.5/weather Auth: appid in query Parameters: lat, lon Endpoints: assistant/session data/2.5/weather data/3.0/onecall	https://api.apilayer.com/currency_data Auth: apikey in header Parameters: source, currencies, amount, from, to, date Endpoints: /live /convert /historical	https://en.wikipedia.org/w/rest.php/v1 Auth: None(Public) orUser-Agent Parameters: q(search),limit(max results) Endpoints: /page/html/ /page/summary/ /search/page	https://api.geoapify.com/v2 Auth: apiKey in query Parameters: categories, filter, bias, limit Endpoints: /places /routing /geocode/search

Frontend Framework:

Tailwind will provide us with much more flexibility to build our own styled elements and components compared to other frameworks, it'll allow us to create much more unique designs for our traveling website

- Gallery: provide image cards that will display country flags, landmarks, etc. Wrapped in gallery component with custom slides per image
- Search form: Provide a card containing text input that will facilitate searching countries, will be reusable and placed both in homepage and country search
- Top-bar: nav bar that provides linkage to user profile, country search, also simple drop-down for currency selection
- Hover effects: pretty-fy our site using built in hover components, ideally wrap it in all buttons that are interactable
- Sign-in/Sign-out: login card, reusable so that it will cover the login/logout pages and also ideally be callable whenever necessary

Task Breakdown:

- API Setup (Mustafa)
  - Get Access Keys, parse data
- HTML Templates Design (Yu)
  - Create Frontend Framework and structure
- Python Flask Scripting (Ricky)
  - Routes to frontend and handles user input
- Python DB and API Scripting (Andrew)
  - Interacts w/ DB and fetches data from API to present to user