



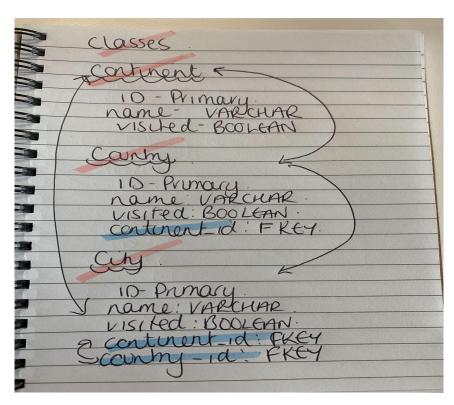
Brief

The app should allow users to create, read, update and delete cities and countries they want to visit. The user had to have the ability to list multiple cities and countries and mark them as wishing to visit, visited or to remove them off their bucket-list.

Concept

The design behind this app involved several different modules. There were three different tables within a database that involved two one to many relationships between the models. Testing was conducted on the models themselves to ensure they were passed. Three different repositories were created alongside three different controllers. The controllers had corresponding HTML files and one set of CSS for the whole application.

Database Diagram



Technical Aspects

This app uses the following:

- → Python3
- → SQL
- → Flask
- → HTML, CSS
- → Jinja
- → Internet Browser such as Chrome

Let's check out Wanderlust!

<form action="/cities" method="post" id="forms"> <div> <label for="city">City:</label> <input type="text" id="city" name="city" placeholder="Paris" required autofocus> </div> <div> <label for="country">Country:</label> <select id="country id" name="country id"> {% for countries in countries %} <option value="{{ countries.id}}">{{countries.name}}</option> {% endfor %} </select> </div> <div> <label for="continent">Continent:</label> <select id="continent" name="continent id"> {% for continents in continents %} <option value="{{ continents.id }}">{{ continents.name }}</option> {% endfor %} </select> </div> <div> <label for="visited">Visited: <select id="visited" name="visited"> <option value="true">Been There Done That! <option value="false">Gotta Save My Dolla To Go!</option>

Pits and Peaks

Pits:

- The models and databases structures
- Confusion around what webpages should look like

Peaks:

- Proud of the menu hover
- Forms hidden functionality

When you realize you're 40,000 feet away from everyone and their crap





