Team FireFox

Type 1 Diabetes Software

main.py

```python
# main.py

from flask import Flask

app = Flask(__name__)

@app.route("/")
def index():
    return "Congratulations, it's a web app!"

if __name__ == "__main__":
    app.run(host="127.0.0.1", port=8080, debug=True)
```

Software Functionality

Our software is a web-based application that allows users to search for data related to single nucleotide polymorphisms (SNPs) on chromosome 6 that are associated with increased risk of Type 1 Diabetes (T1D).

Software Functionality

The webpage takes in three types of entries in its search query : an rsID, a gene name or a location range so the user can retrieve relevant information for T1D on chromosome 6.

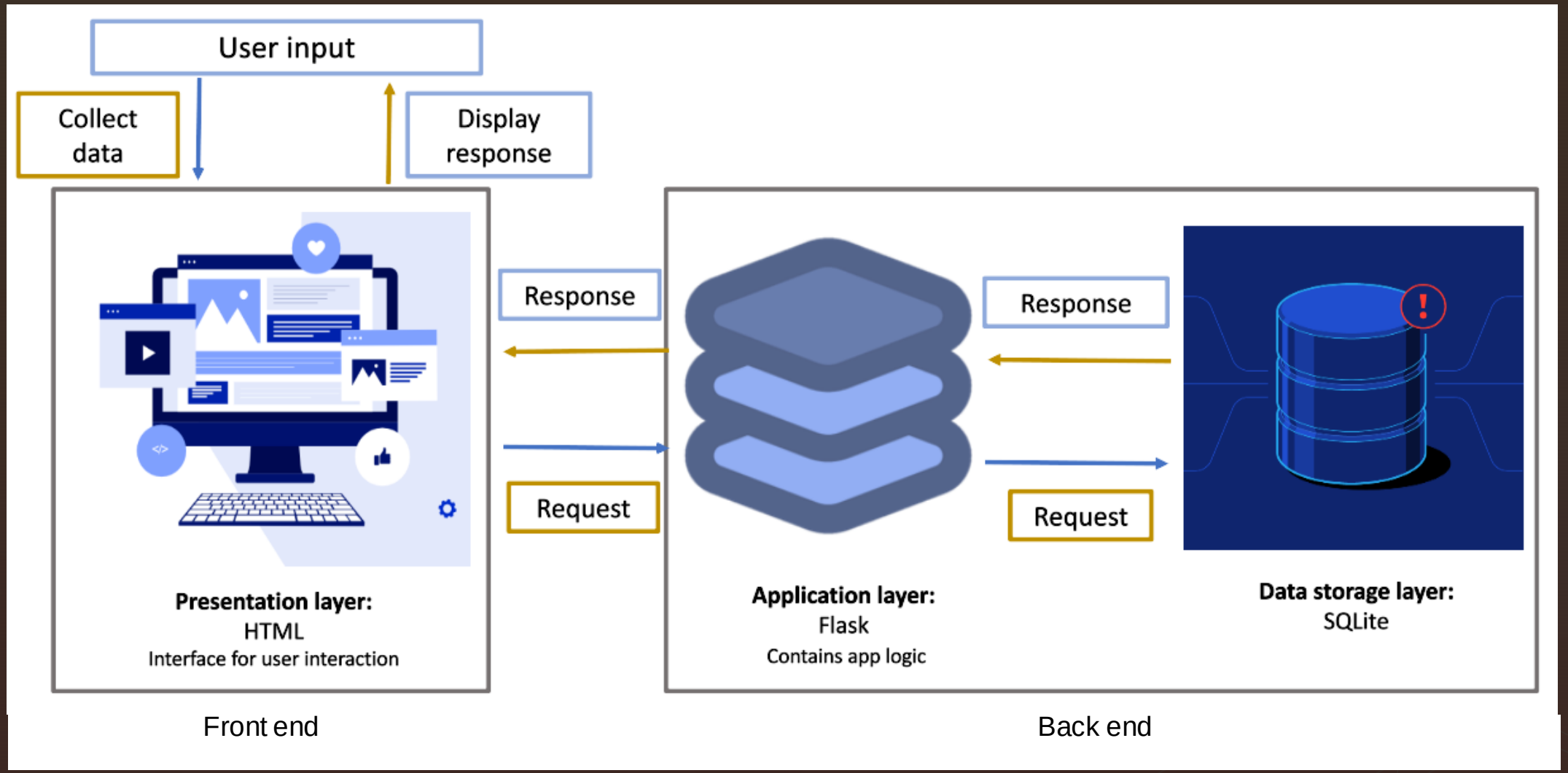
Software Functionality

- If a single SNP is returned from the search, the user will be able to see the following attributes,
- SNP name, genomic position, p-value from the association test, mapped gene name, variant frequency in three different human populations of interest, clinical relevance for each variant, and one gene ontology term associated with each mapped gene.

Software Functionality

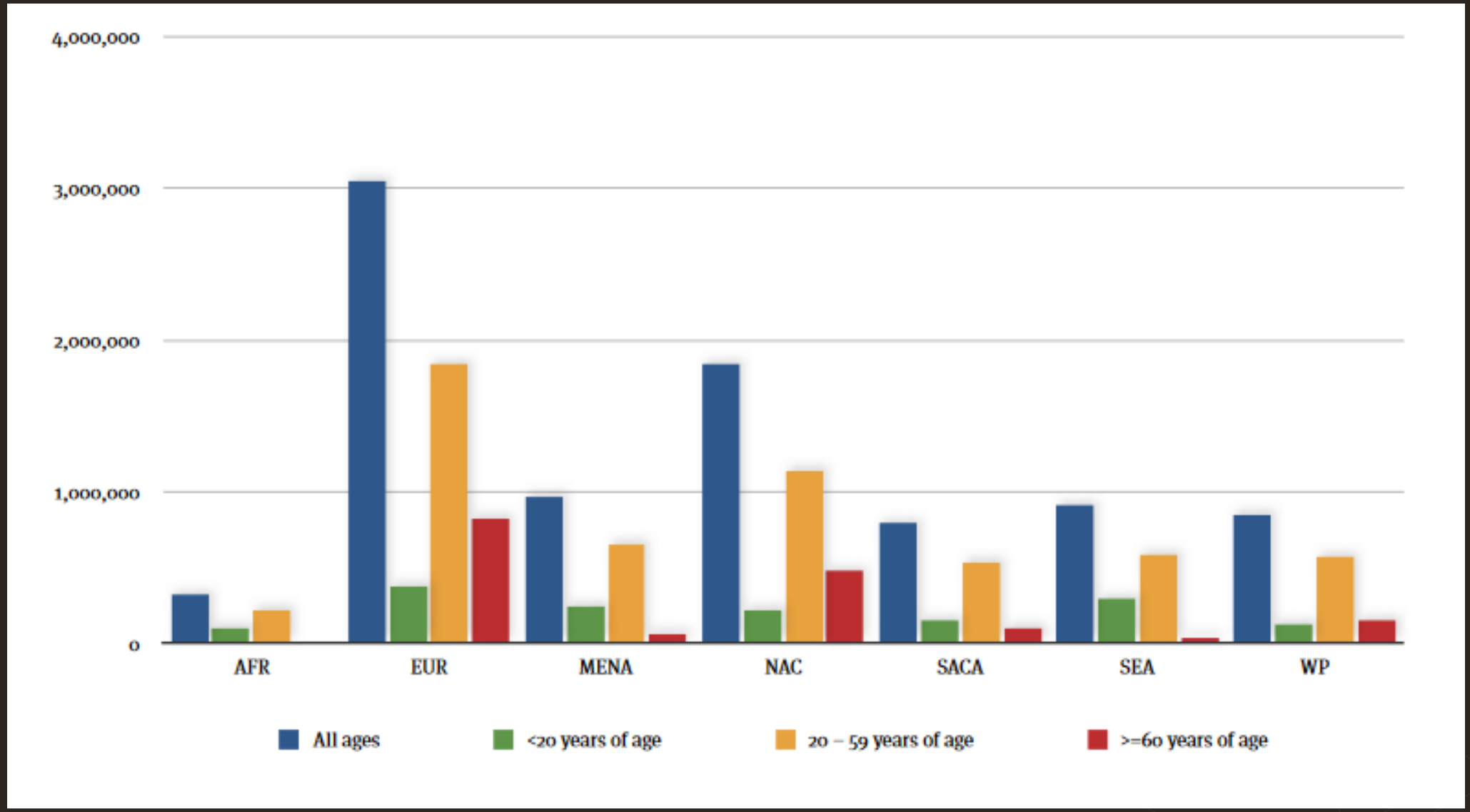
- For multiple SNP's all the attributes mentioned previously are returned, as well as, a Manhattan Plot, Linkage Disequilibrium (LD) scores, and LD plot, and a downloadable CSV file of the LD score for each human population of interest.

Architecture



Data Sources

- ***GWAS Catalog -*** SNPs associated with T1D on chromosome 6 and their relevant information necessary for the software were found in the GWAS catalog.
- ***CADD -*** Combined Annotation Dependent Depletion, is a computational tool used to predict the functional impact of SNPs.
- ***Variant clinical relevance -*** categorised the scaled CADD scores obtained for each SNP Risk allele frequency.
- ***Risk Allele Frequency -*** three populations chosen: one with a high prevalence of the disease, one with a medium prevalence and one with a low prevalence.
- ***GO terms -*** provide a link between a gene product and a biological process, cellular component, or molecular function.

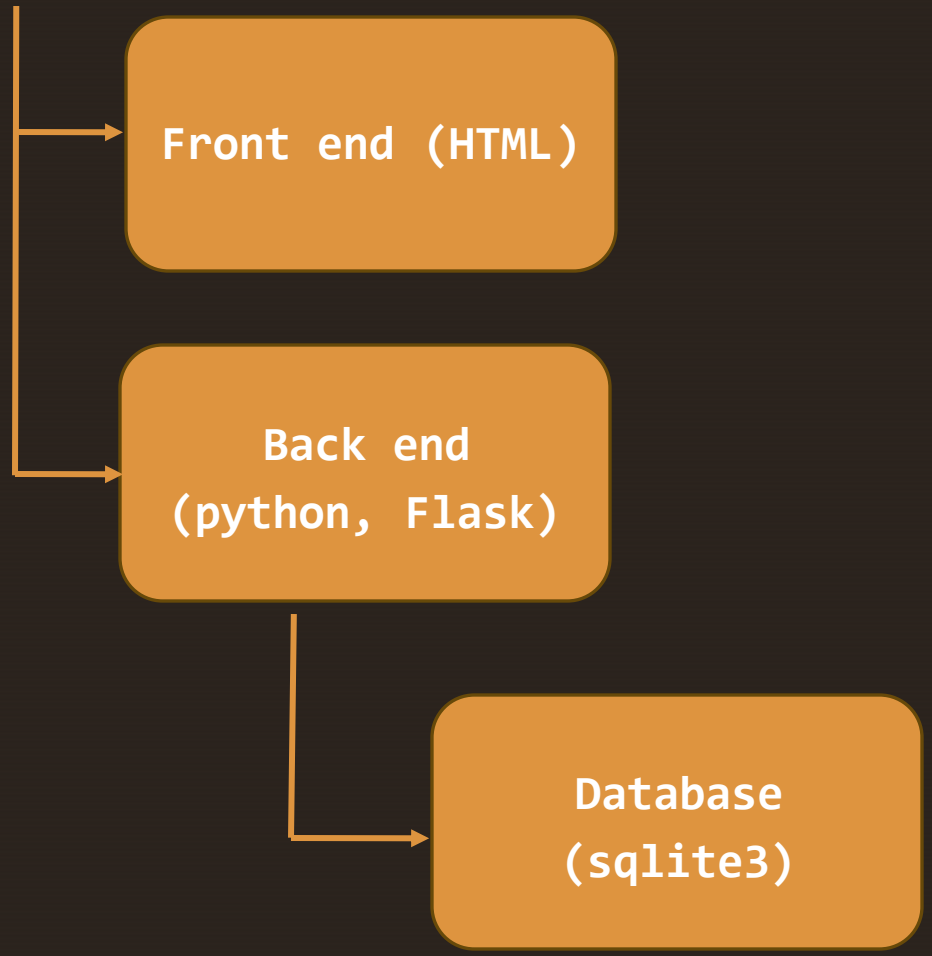

4,000,000
3,000,000
2,000,000
1,000,000
0
AFR
EUR
MENA
NAC
SACA
SEA
WP
All ages
<20 years of age
20 – 59 years of age
>=60 years of age

Data Sources

- ***GWAS Catalog -*** SNPs associated with T1D on chromosome 6 and their relevant information necessary for the software were found in the GWAS catalog.
- ***CADD -*** Combined Annotation Dependent Depletion, is a computational tool used to predict the functional impact of SNPs.
- ***Variant clinical relevance -*** categorised the scaled CADD scores obtained for each SNP Risk allele frequency.
- ***Risk Allele Frequency -*** three populations chosen: one with a high prevalence of the disease, one with a medium prevalence and one with a low prevalence.
- ***GO terms -*** provide a link between a gene product and a biological process, cellular component, or molecular function.

Software Technologies

- Consist of 3 components:




```html
<html>
  <head>
    <title>Team Firefox - Type 1 Diabetes Application</title>
  </head>
  <body>
    <h1 style="color: teal;font-family: Calibri">Team Firefox - Type 1 Diabetes Data Application</h1>
    <form method="POST">
      {{ gform.hidden_tag() }}
      <p>
        <label style="font-family: Calibri;">{{ gform.gene_name.label }}</label>
        {{ gform.gene_name(size=32) }}
        {% if gform.gene_name.errors %}
```

```python
from flask_wtf import FlaskForm
from wtforms import StringField, SubmitField
from wtforms.validators import InputRequired

import requests
import re
import numpy as np
from ld_plot.ld_plot import ld_plot
import matplotlib.pyplot as plt
import itertools

app = Flask(__name__)
app.config['SECRET_KEY'] = 'Batman:D'
```

```python

con = sqlite3.connect("app_data_comp.db", check_same_thread=False)
cur = con.cursor()
con.row_factory = sqlite3.Row

class rsIDForm(FlaskForm):
    rs_ID = StringField('Enter a valid rsID value:', validators=[InputRequired()])
    submit_rsID = SubmitField('Search')
```

Software Technologies

- Original csv files containing data used to generate database
- Python preferred programming language
- Used to build the web app from the ground up
- ***SQlite3*** – provides lightweight disk-based database
  - Used for prototype web application building
  - Queries to SQL databases can be run with Python
  - Combined with libraries such as pandas make for great data manipultion

In [1]:
```
# Import packages
from pathlib import Path
import sqlite3
import pandas as pd
```

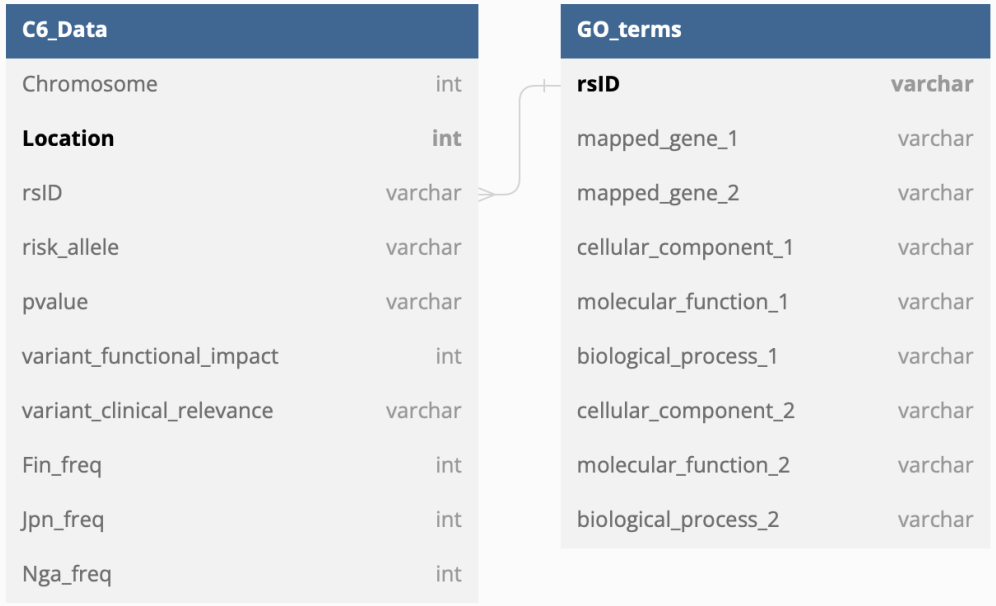
In [2]:
```
# Create empty database
Path('app_data.db').touch()
```

In [3]:
```
# Connect to database and create cursor
conn = sqlite3.connect('app_data.db')
cur = conn.cursor()
```

In [4]:
```
# Load the chromosome 6 data into a Pandas DataFrame
Chromosome_6_data = pd.read_csv('Chromosome_6_data.csv', encoding= 'unicode_escape')

# Check data has loaded correctly - remove from final code!!
Chromosome_6_data.head()
```

	rsID	risk_allele	pvalue	RAF	mapped_gene	reported_trait	traits	location
0	rs11755527	G	3 x 10-8	None	BACH2	Type 1 diabetes	type 1 diabetes mellitus	6:90248512
1	rs11755527	G	5 x 10-12	0.47	BACH2	Type 1 diabetes	type 1 diabetes mellitus	6:90248512
2	rs9272346	G	6 x 10-129	None	HLA-DQA1	Type 1 diabetes	type 1 diabetes mellitus	6:32636595
3	rs78824139	C	9 x 10-6	0.986	EYS	Type 1 diabetes (age at diagnosis)	age at diagnosis, type 1 diabetes mellitus	6:64748694
4	rs114631266	C	2 x 10-6	0.974	OR2P1P, OR2W1	Type 1 diabetes (age at diagnosis)	age at diagnosis, type 1 diabetes mellitus	6:29058810

In [5]:
```
# Write the data to a SQLite table
Chromosome_6_data.to_sql('C6_data', conn, index = False)
```



Database Schema

- The code written to create the database, makes an empty database file named **app_data.db**
- Connects to it, making two tables named **C6_data** and **GO_terms**
- Populates these tables with data read from their corresponding CSV files.

- The **primary key** for **C6_data** is L**ocation**, and the **primary key** for GO_terms is **rsID**. These two tables are linked by **rsID**, with this being the **foreign key** for **C6_data**.

Software Technologies

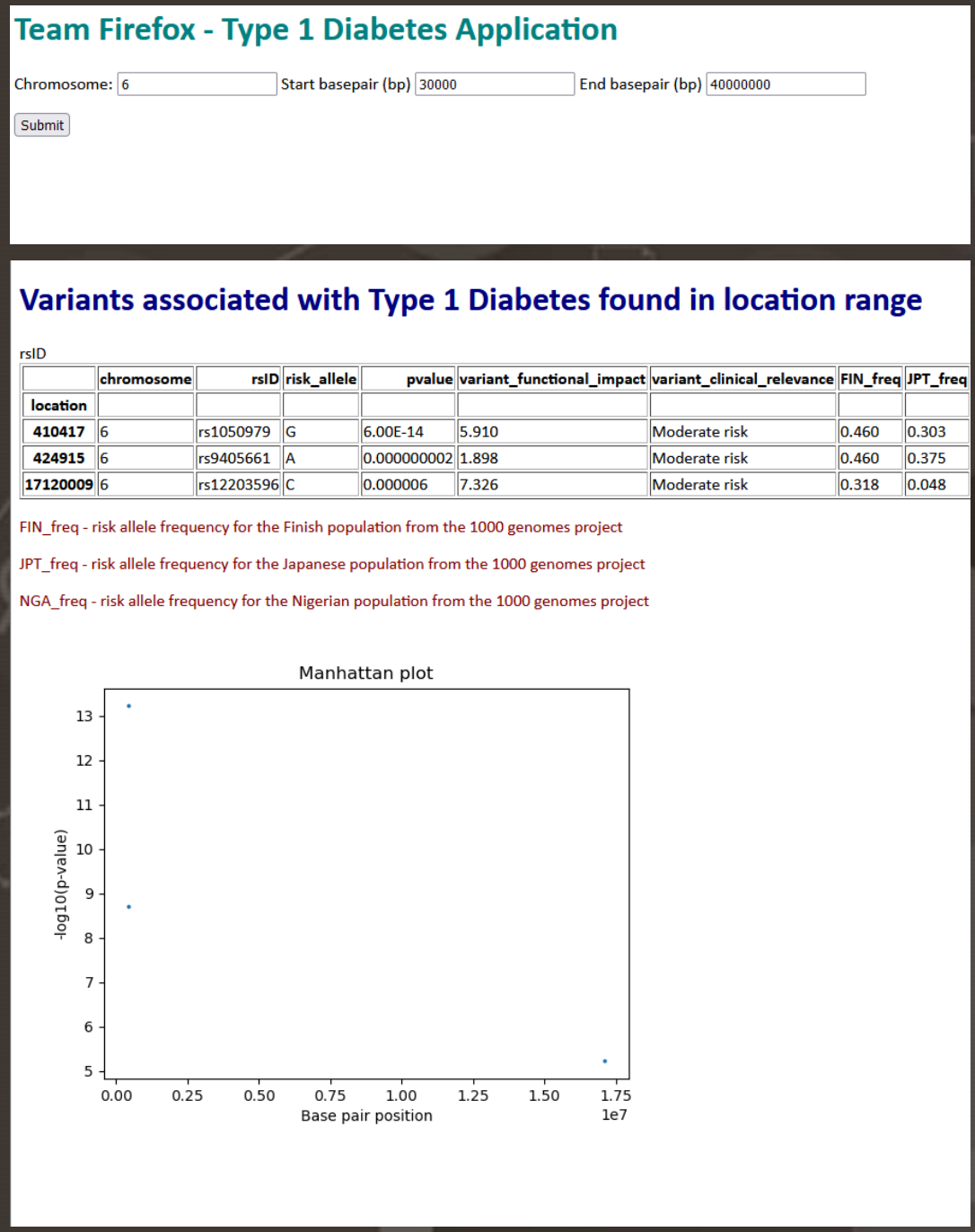
- ***Flask*** – A micro web framework lightweight and flexible
- Versatile and independent, user needs to import specific libraries
- Making it easy to build and customize the web application
- ***However:***
  - Important to work in a virtual environment
  - Installing specific libraries and versions may contradict pre-existing ones

```
# Importnat necessary packges
from flask import Flask, render_template, url_for, redirect, request, make_response
import pandas as pd
import sqlite3
import io
import base64
from matplotlib.backends.backend_agg import FigureCanvasAgg as FigureCanvas
import matplotlib.pyplot as plt
import urllib.request

from flask_wtf import FlaskForm
from wtforms import StringField, SubmitField
from wtforms.validators import InputRequired

import requests
import re
import numpy as np
from ld_plot.ld_plot import ld_plot
import matplotlib.pyplot as plt
import itertools
```

```
@app.route('/')
def index():
    return render_template('base_index.html')

# Define action for SNP route to retrieve data based on rsID input
@app.route('/SNP', methods=['GET', 'POST'])
def SNP():
    # Assign response of rsIDForm to variable
    rform = rsIDForm()

    # Validation of user input and data extraction
    if rform.validate_on_submit():

        # Get rs_ID value from form data
        rs_ID = rform.rs_ID.data.lower()

        # Query the database
        data = pd.read_sql_query('''SELECT C6_data.rsID, location, risk_allele, pvalue, variant_functional_impact, \
               variant_clinical_relevance, FIN_freq, JPT_freq, NGA_freq, mapped_gene_1, mapped_gene_2, \
               biological_process_1, cellular_component_1, molecular_function_1, biological_process_2, cellular_component_2, \
               molecular_function_2 FROM C6_data, GO_terms WHERE C6_data.rsID = GO_terms.rsID''', con, index_col='rsID')

        # Convert database query output to dataframe
        df = pd.DataFrame(data, columns=["location", "rsID", "risk_allele", "pvalue", "variant_functional_impact", \
                "variant_clinical_relevance", "FIN_freq", "JPT_freq", "NGA_freq", "mapped_gene_1", "mapped_gene_2", \
                "biological_process_1", "cellular_component_1", "molecular_function_1", "biological_process_2", "cellular_component_2", \
                "molecular_function_2"])
```

Software Technologies

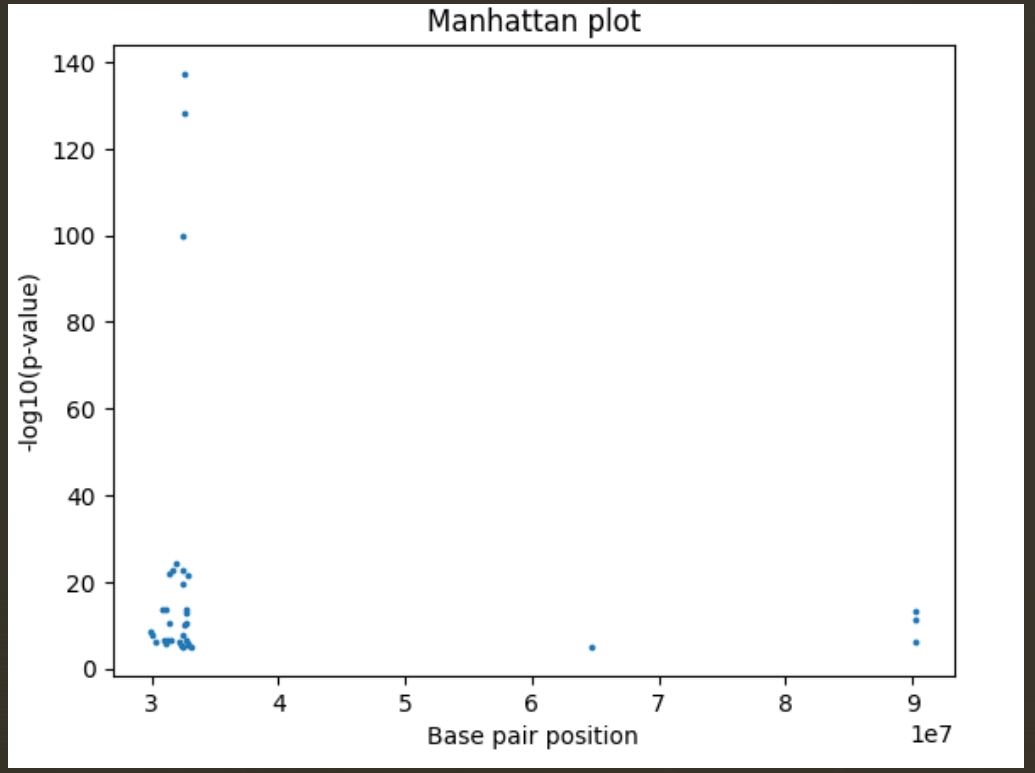
- ***HTML - T***o render the user interface
- Allow for a more aesthetically pleasing interface
- More interactive experience
  - Makes data manipulation and extraction easier
  - Allows for creating elements to ouput plots
  - Tables
  - Functions for downloading data

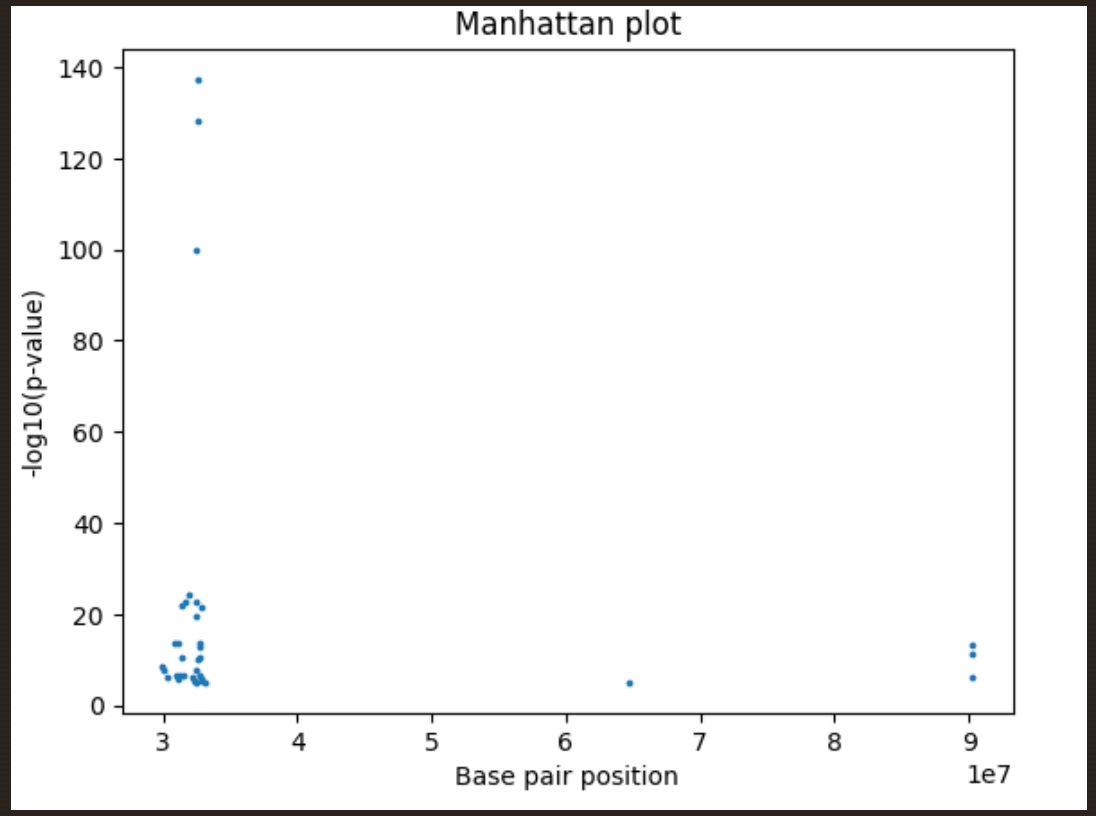


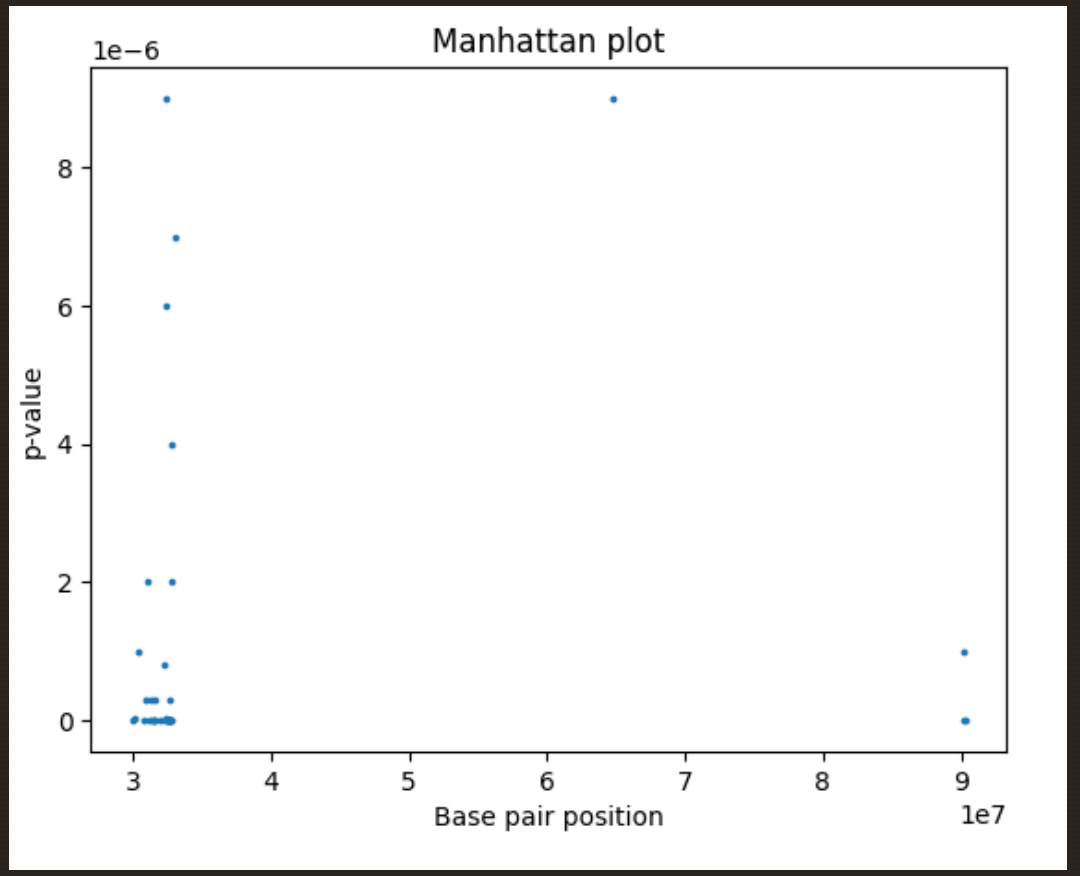
Summary Statistics Methods

Manhattan Plot: Script was written within in the Location function and generates the Manhattan plot of negative logarithm of p-values for all T1D associated SNPs located within a user specified genomic coordinate range.

The plot helps to identify significant genomic regions and potential candidate genes that may be involved in the trait of interest.




Manhattan plot
-log10(p-value)
140
120
100
80
60
40
20
0
3
4
5
6
7
8
9
Base pair position
1e7

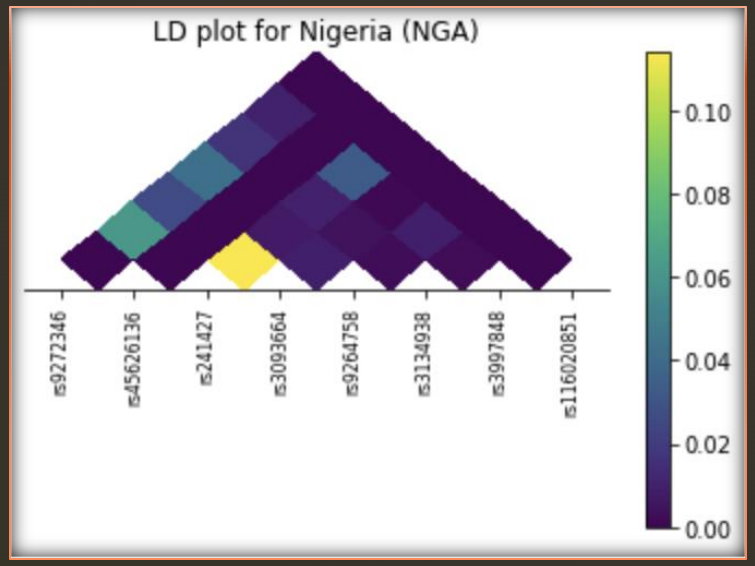

Manhattan plot
1e−6
p-value
8
6
4
2
0
3
4
5
6
7
8
9
Base pair position
1e7

Summary Statistics Methods

Linkage Disequilibrium – Used API from LDpop website to access R^2 values for each SNP.

Data then wrangled into a data frame for download by user and into a symmetrical matrix for input into ld-plot function

Plotted using LD-Plot package.



Example LD plot for Nigeria Population

	rs9272346	rs45626136	rs241427	rs3093664	rs9264758	rs3134938	\
rs9272346	1.0	0.0021	0.0101	0.0014	0.0094	0.0003	
rs45626136	0.0021	1.0	0.0165	0.0009	0.0001	0.0025	
rs241427	0.0101	0.0165	1.0	0.0005	0.0287	0.0181	
rs3093664	0.0014	0.0009	0.0005	1.0	0.0032	0.0031	
rs9264758	0.0094	0.0001	0.0287	0.0032	1.0	0.0816	
rs3134938	0.0003	0.0025	0.0181	0.0031	0.0816	1.0	
rs3997848	0.002	0.0025	0.0919	0.005	0.0098	0.004	
rs116020851	0.0018	0.0009	0.0036	0.0002	0.0032	0.0031	

Example of Symmetric Matrix

Limitations and Future Development

- Lack of information for some SNPs
- Excessive navigation pathway length for Linkage Disequilibrium
- Reduced error handling capabilities
- Aesthetically simple design
- Linkage disequilibrium calculator can be slow
- Only provides data for Chromosome 6 SNPs
- Local software

Software Demonstration

We will now walk you through a practical demonstration of our complete software ...