**Section 1. Web App Architecture**

1. Data Storage

We are using a SQlight database so we are able to use this stand-along with the application deployment. Django allows us to interact with the database through python abstraction. Here are some steps that detail how this works.

* By defining models you create a table in the database and each field defined in the model will be a column in the database.
* You create the tables using djangos built in ```manage.py makemigrations``` command then ```migrate```. These commands create the tables based on the models.
* To perform CRUD operations you use Django’s Object-Relational Mapping (ORM) API.
* Django generates SQL SELECT statements and retrieves data from the database when using the ORM.
* Data retrieval is accomplished by using Django’s QuerySet API. Ex. (Renewables.objects.all())

1. Back-End

Django sites are primary developed with python. However it relies on SQL to interact with the database. HTML, CSS, and JavaScript can be used on the front end.

1. Database Access & Security

Django will allow us to connect to the database with its built-in framework. Django provides an ORM framework that abstracts away the details of accessing the database, allowing you to perform database operations using python code instead of SQL queries.

Django provides multiple built-in features to help with data security, here are two main ones;

* Allows the use of a secure connection, HTTPS or SSL.
* Allows you to limit database access to only grant access to users and applications that need it.

We will set up users and admin is a built in function of Django that we will use.

Users will only be able to query the database using defined fields in the application, they will not have access to view or modify any data outside of the provided graphic by default. We are not capturing any new data from the user.

1. Front-End

We will be using Bootstrap. HTML pages in Django allow us to imbed python code along with the ability to extend Bootstrap front end CSS. This will give us access to things like crispy forms and other Bootstrap built in functionality that will make front end development easier.

1. Deployment

The application will be hosted on PythonAnywhere by Anaconda, which is a web-based server management service for developing with Python. Some benefits of using Puthoneverywhere are that it is easy to use, provides a pre-configured Python environment, manages server, allows one free project, provides a pre-configured Django environment, web-based file editor, git integration, and of course good documentation!

1. Interactivity

We are creating an interactive chart. This app will allow a user to view the actual load data for a user input timeframe, then allow for the user to add various forms of energy generation to see the generation profile make up of that given source. We will have a drop-down button to change from fossil fuels to wind, solar, hydro, etc. This data will use the user time input as well to change the chart.

1. Diagram

   Description automatically generatedWeb Application Architecture

#### **Section 2. Web App Layout**

* What is the initial layout (when a user sees your app first)?
* Where is the menu panel?
* How many pages do you need? Or will you be using Tabs?
* What is the color schema?
* What each page or Tab will display?
* What functionalities will be available and how users will access them (e.g. search/query box/drop menu ...)

The layout for our site will have three pages. First there will be a log in page for users to sign into the site. Then it will direct them to our home page upon login. On the first page we will have two parameters for a user to enter, a date range (within the dataset) and a drop-down list of energy generation sources. Once filled out it will display on a chart the x axis will have the date range the user entered and the y axis will display the actual load over that timeframe along with whatever the user chose for their energy generation source. This will show how much of the energy needed over that timeframe was generated by the selected generation method. There will also be an about page that you can click on that will be an explanation of the data and how to make best use of the chart.

There is a app mock up created to show how the app will work. (This was done using Pencil website/application mock up software.) Below are screenshots of the HTML file created.

This is the home page. Once logged in you can see this page that allows you to enter your date range and your energy generation type, hit the start button, and it will plot on the page.

Graphical user interface, application

Description automatically generated

Below is the about page. This will allow for us to describe the project details and provide information about the data being used. Additionally on both pages you see a link to the user portal. For the admin account that is how you access back-end permissions. This is on both the home and about pages.

Graphical user interface, text, application

Description automatically generated

#### **Section 3. Individual and Team Work Assessment - 10pts**

* If you are working individually, please describe your work: [Are you satisfied with the task completion (scale 1-10),  time commitment, what could be done better
* Teams:
  + Describe your personal satisfaction with the task: [Are you satisfied with the task completion (scale 1-10), time commitment, and what could be done better?
    - We need to make sure everyone is participating and work is equally distributed. While the grade is initially provided for the entire team, we reserve the right to change a grade for any individual students who did not fully participate and fulfill their obligations/responsibilities in the project.
  + Let us know if someone is not responding/participating - we will reassign that person to an individual group