**Work List Application**

The Work List Application is designed and implemented through Django and Python by providing a complete User Interface for the user that is easily accessible and simple to use; this application aims to provide a method to note down activities, tasks, or any work that will be done throughout the day. It is meant to provide a means by which tasks can be set at the start of the day, then appropriately edited or removed once something is completed. In case a job was not completed or is pending, there is a cross out method which is meant to represent tasks that are not met for the day and are to be handed over to the next day; once a job is completed, it should be deleted, since the aim of the application is a daily task co-ordination method.

First, to use the application, start in the same repository as the manage.py file. Then within the terminal, use the following command: <pip install -r requirements.txt> This command installs the required packages. Then use the command: <python manage.py runserver>. This command runs the server for the application. Afterwards, the application can be accessed with the URL: http://127.0.0.1:8000/

Adding a task to the list is done by selecting the Taskbar and printing out the task that should be done, then clicking add to list. There are Edit and Delete buttons for the job, alongside a cross-off Button to represent in-complete tasks.

Diagram

Description automatically generatedFigure 1. represents the flow char that showcases; the Insertion of the data. The User Input is the data being inserted which can be something like; “Cooking Dinner”, then the method will be checked; if the method is a ‘POST’ method, that will imply insertion. That information is then checked if the form of the method is valid. The item is added to the database. Afterwards, the database will be exported as an output to the HTML that represents the list being shown.

Figure

A screenshot of a computer

Description automatically generated with medium confidenceFigure 2. represents the data insertion part; the first IF statement checks which type of request is required. The second IF statement checks if the inserted information is empty and redirects the user home. The Item variable is created, and the list within the database is checked. The database in this case being used is sqlite3. Then finally redirects to the Home page while showing the updated list.

Figure