SOFTWARE ENGINEERING LAB PROJECT 2020

DURANGO

FRONT-END/GUI, BACKEND & DATABASE

SURYANSH KUMAR
18CS30043

FRONTEND / GUI:

- **HTML** and **CSS** form the basic structure of our application's frontend.
- **Bootstrap** classes are used for elements like buttons and flash messages.
- **Jinja2** (template engine used to render web templates in Flask) is used for template inheritance. A layout page with a common background and navigation bar is inherited by all webpages.
- **Javascript** is used for the typewriter effect on the home page, in pie charts, to show error messages, create toggle effects in buttons and the chat UI..
- **AJAX** call is used to hide/show the mail body in the mail_tasks page.

BACKEND:

- The **routes.py** file forms the backend skeleton, consisting of all decorators and paths.
- **FLASK-SQLALCHEMY** is used as the Object Relational Mapper (ORM) acting as the interface between our python script and database. It helps in realising the user and task models (in **models.py**) as tables in our database.
- **FlaskForm** is used to create the structure of a form. I have used **WTForms** for form validation. All form classes are included in **forms.py**. **Flask-Bcrypt** is used to hash account passwords before storing them in the database.
- **Twilio** and **SMS4IND** APIs are used for sending OTPs and sms reminders respectively.
- **search.py** consists of the KMP search algorithm code used in search bar and **mails.py,** the latter being used to extract 'important' mails.
- **Flask-Socket** is used to wrap the app in the Socket IO functionality, enabling a real time chat.

DATABASE:

- SQLite3, a **relational database** is used to store all data. The ORM facilitates creation of the database with a single command from the terminal.
- The database consists of **three** tables: **User**(with 8 columns) ,**Task**(with 10 columns) and **Connections**(with 4 columns). The user_id column is the **primary key** for the user table and **foreign key** for the Task table. It is used to map tasks with users. Two columns of Connections table(user_a, user_b) have a relationship (it is specified how to handle multiple join paths in the square brackets of foreign_keys) with the User table,