PROJECT REPORT ON SUICIDE PREDICTION AND MENTAL HEALTH ANALYSIS

SUBMITTED BY:

NAG ASHISH S V	(221003064)
PRAMODH SAIRAM PV	(221003117)

PROBLEM STATEMENT :

Mental health includes our emotional, psychological, and social well-being. It affects how we think, feel, and act. It also helps determine how we handle stress, relate to others, and make choices. Mental health is important at every stage of life, from childhood and adolescence through adulthood.

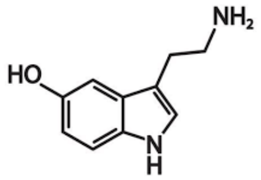
Over the course of your life, if you experience mental health problems, your thinking, mood, and behaviour could be affected. Many factors contribute to mental health problems, including:

- Biological factors, such as genes or Molecular chemistry of brain
- Life experiences, such as trauma or abuse
- Family history of mental health problems
- Extensive Drug abuse and many other addictions

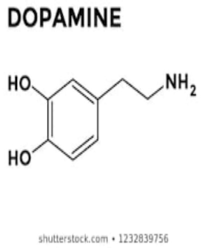
Thus this "**Suicide Prediction and Mental Health Analysis**" system will be a appropriate monitor for the mental health of a person .Here we consider the Neuro chemical analysis inorder to find the patterns that trigger the suicidial thoughts in person

The birth of the idea of suicide in a person is in-fact sometimes based upon the presence of five certain chemicals in a particular proportion .

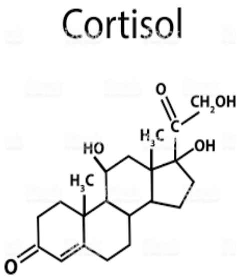
These chemicals are :



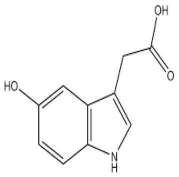
1) Serotinin



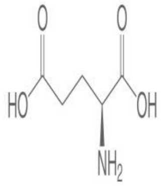
2) Dopamine



3) Cortisol



4) 5-HIAA



5) Glutamte

There is also a minimum influence of other chemicals such as Norepinephrine, Adrenalin

Manual calculations and conversion for each and every person is tough and time consuming thus we developed a **Machine Learning model** which takes in the amount of the above five Neuro chemicals in a human body and will predict the percentage of how prone a person is to commit suicide

AIM & OBJECTIVE :

The purpose of this document is to present a detailed information about the "**Suicide Prediction and Mental Health Analysis"** system

It will explain the purpose and features of the software . This document is intended for users of the software and also potential developers

Through some research papers we found there is pattern behind the proportions of this chemicals in the persons who commited suicide and who are about to commit. Interestingly a chemical called "Serotonin (C1=CC2=C(C=C1O)C(=CN2)CCN)" plays a crucial role behind their thoughts and ideas that triggers the depression and suicidial thoughts, along with other chemicals like dopamine($C_8H_{11}NO_2$) and glutamate($C_5H_9NO_4$) . See the ref below

https://www.researchgate.net/publication/23238163_The_role_of_dopamine_and_serotonin_in_suicidal_behaviour_and_aggression

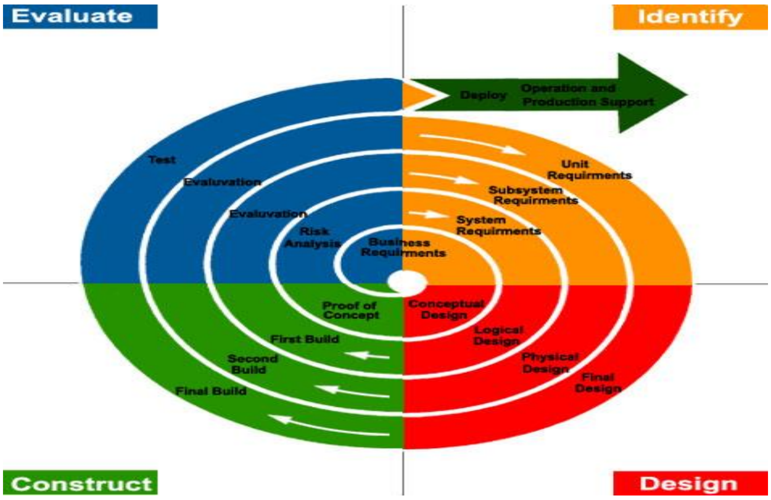
The main objective of this system is to diagnose the mental health and %thoughts of suicide for a person at the right time based on the levels of those 5 chemicals in their body, prevent suicide and give him suggestions of necessary psychiatrists and mental health clinics around his place .

PROCESS MODEL:

Choice : Spiral Model

Spiral Model is a combination of a waterfall model and iterative model. Each phase in spiral model begins with a design goal and ends with the client reviewing the progress. The spiral model was first mentioned by Barry Boehm in his 1986 paper.

The development team in Spiral-SDLC model starts with a small set of requirement and goes through each development phase for those set of requirements. The software engineering team adds functionality for the additional requirement in every-increasing spirals until the application is ready for the production phase.



Identification:

This phase starts with gathering the business requirements in the baseline spiral. In the subsequent spirals as the product matures, identification of system requirements, subsystem requirements and unit requirements are all done in this phase. This phase also includes understanding the system requirements by continuous communication between the customer and the system analyst. At the end of the spiral, the product is deployed in the identified market.

Design:

The Design phase starts with the conceptual design in the baseline spiral and involves architectural design, logical design of modules, physical product design and the final design in the subsequent spirals

Construct-Build :

The Construct phase refers to production of the actual software product at every spiral. In the baseline spiral, when the product is just thought of and the design is being developed a POC (Proof of Concept) is developed in this phase to get customer feedback.

Then in the subsequent spirals with higher clarity on requirements and design details a working model of the software called build is produced with a version number. These builds are sent to the customer for feedback.

Evaluation and Risk analysis:

Risk Analysis includes identifying, estimating and monitoring the technical feasibility and management risks, such as schedule slippage and cost overrun. After testing the build, at the end of first iteration, the customer evaluates the software and provides feedback

Reasons for choosing Spiral Model:

- When there is a budget constraint and risk evaluation is important.
- For medium to high-risk projects.
- Long-term project commitment because of potential changes to economic priorities as the requirements change with time.
- Customer is not sure of their requirements which is usually the case.
- Requirements are complex and need evaluation to get clarity.
- New product line which should be released in phases to get enough customer feedback.
- Significant changes are expected in the product during the development cycle.

BENIFITS OF SPIRAL MODEL:

- Changing requirements can be accommodated.
- Allows extensive use of prototypes.
- Requirements can be captured more accurately.
- Users see the system early.
- Development can be divided into smaller parts and the risky parts can be developed earlier which helps in better risk management.

EXISTING SOFTWARE:

Many papers have been published in the prevention of suicide using Machine Learning in the domain of psychology and psychology. see the ref

https://www.researchgate.net/publication/330156470_Prediction_of_Suicide_Causes_in_India_using_Machine_Learning

https://www.researchgate.net/publication/328225907_Use_of_a_Machine_Learning_Algorithm_to_Predict_Individuals_with_Suicide_Ideation_in_the_General_Population.

LIMITATIONS OF EXISTING SOFTWARES:

- There is no software that was built exclusively to predict the percentage of suicidial thoughts in human based on Neuro chemicals and Neuro transmitters.
- Most of the mental health analyzing softwares deal with social physiological and pshychological parameters to estimate the amount of mental illness.
- The performance of the existing softwares doesn’t consider the chemical levels of the brain which leads to less accuracy of the results.

NOVELTY OF THIS APPLICATION:

we have done the prediction analysis in the **neurology domain and molecular chemistry of brain** which influences the suicidial thougths, by considering the following references.

https://www.researchgate.net/publication/329660527_Serotonin_receptors_and_suicide_major_depression_alcohol_use_disorder_and_reported_early_life_adversity

https://psychiatryonline.org/doi/pdf/10.1176/appi.ajp-rj.2017.120105

LIMITATIONS OF THE APPLICATION:

The main limitation is that we have to take a few tests physically to analyse the amount of these chemicals in our body which in-turn will be used as the input for the model.

The lack of availability of these chemical datasets online which leads to slow research and further developments.

METHODOLOGY-MODULES & FUNCTIONALITIES

Understanding about the chemicals that plays a major functionality in prediction

1. **Serotonin:**
   - Normal levels: 101-283 ngms/ms
   - Low levels: simulate suicidial thoughts
   - High levels: carcinoid syndrome
   - Effects: appetite arousal mood, associated with sleeping eating and digestion, In normal levels, happy calm more focussed less anxious, more emotionally stable, In low levels, lead to insomnia and suicidial thoughts

2. **Dopamine:**
   - Normal levels: 195.8 pmol/ltr (approx)
   - Effects: behaviour, emotion cognition etc
   - Low levels: suicidial thoughts

3. **Glutamate:**
   - Normal levels: 50-100 micro mol/litre
   - Effects: cognition, learning, memory
   - less levels : suicidial thoughts

4. **Norepinephrine:**
   - (very least correlated to commit suicide)
   - normal level: 413.8-10048.7 pmol/L
   - Effects: anger and fight simulating feeling, blood pressure
   - low levels : suicidial thoughts

5. **Cortisol:**
   - normal levels: 6-23 ng/ml.
   - high levels: suicidial thoughts.
   - Effects: stressed, drugs intake leads to abnormal levels of cortisol.

6. **5-HIAA:**
   - normal range: (10.5-36.6 micromol/24h).
   - less levels will trigger suicidial thoughts.

Other chemicals like adrenalin etc also plays a key role in mental health. The application predicts by analyzing the pattern followed by these chemicals that triggers suicidial thoughts

MODULES AND FUNCTIONALITIES:

FUNCTIONAL REQUIREMENTS:

This is a single page web application made using following modules

1) **Login Module:** Users can login using their email and password

2) **Registration Module:** New users can register using their details and other identity proofs like health card number.

3) **Analysis Module:** Users can enter their report values of chemicals to check how much prone they are towards commiting suicide.

4) **Report Module:** The report gets generated dynamically along with appropriate predictions made by the ML Module.

5) **Suggestions Module:** Suggestions are generated dynamically based on the report for each user individually

6) **Recommendations Module:** Appropriate recommendations are made based on the user's status and recommends doctors and hospitals

NON-FUNCTIONAL REQUIREMENTS:

1) **Usability**: The application is used by a doctor/any user who want to check mental health status of a person based on already evaluated Neuro chemical analysis report.

2) **Reliability**: The application uses a cloud based database which makes the data more secured and reliable. The document database also provides the support of API to retrieve JSON documents

3) **Supportability**: The system is well built to support any machine. Maintainbility of the system is also easy.

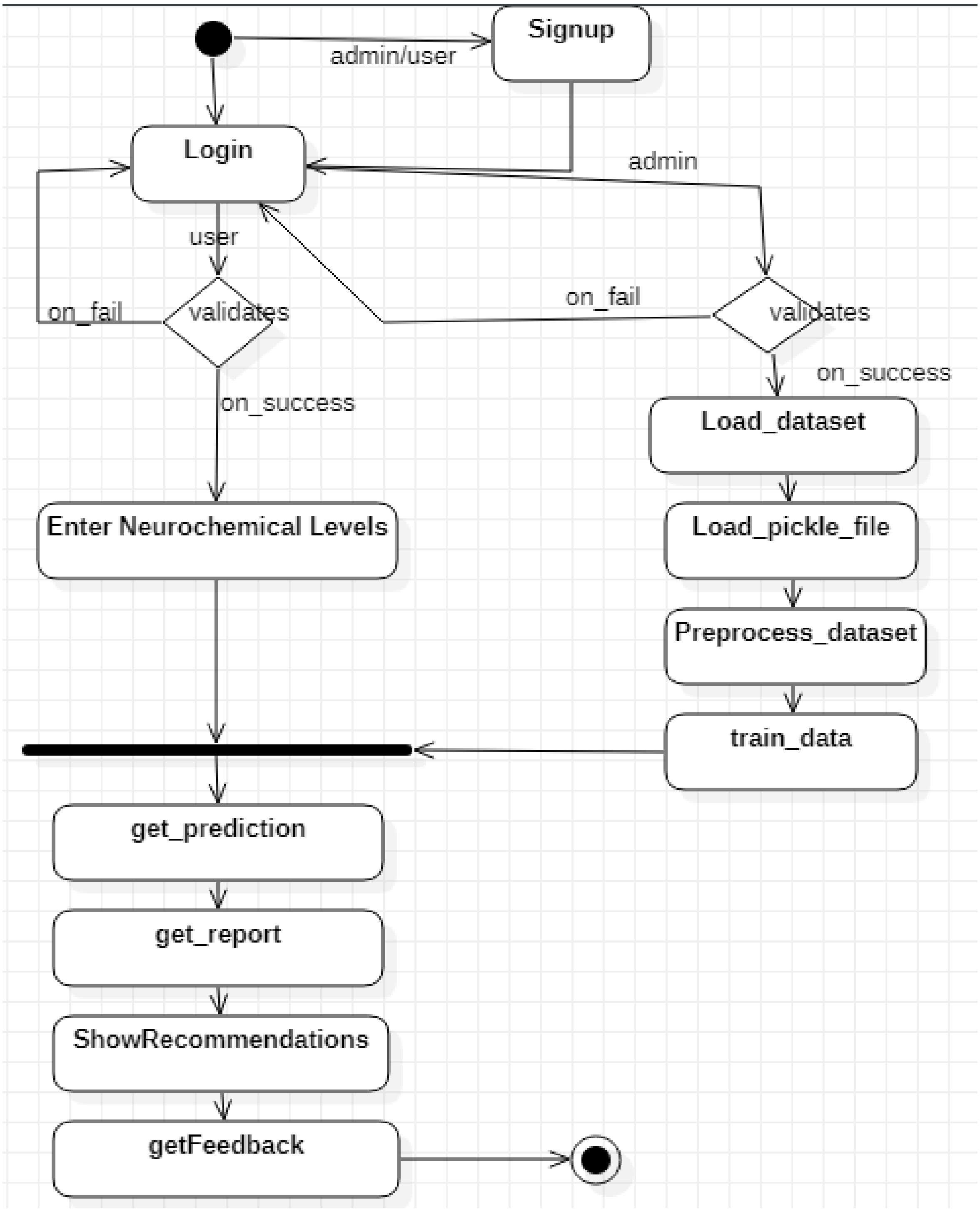
4) **Performance:** The ML model is trained and dumped into a pickle file before the user makes request which avoids redundant pre-processing and training of the ML model, which boosts the performance and reduce latency.

5) **Availability:** The application is deployed on Amazon EC2 instance and uses a cloud database which is available 24*7

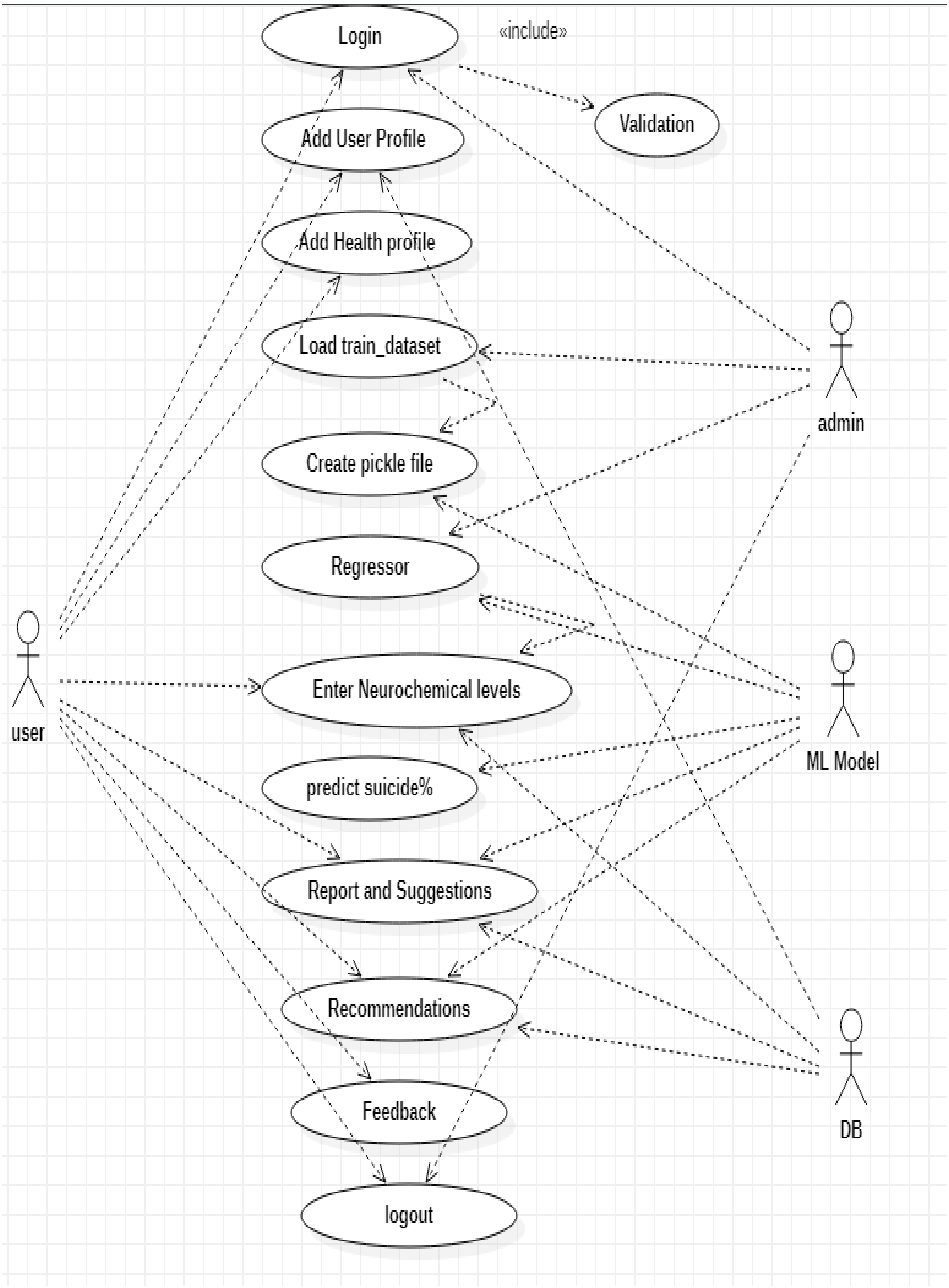
DESIGN ENGINEERING

UML DAIGRAMS:

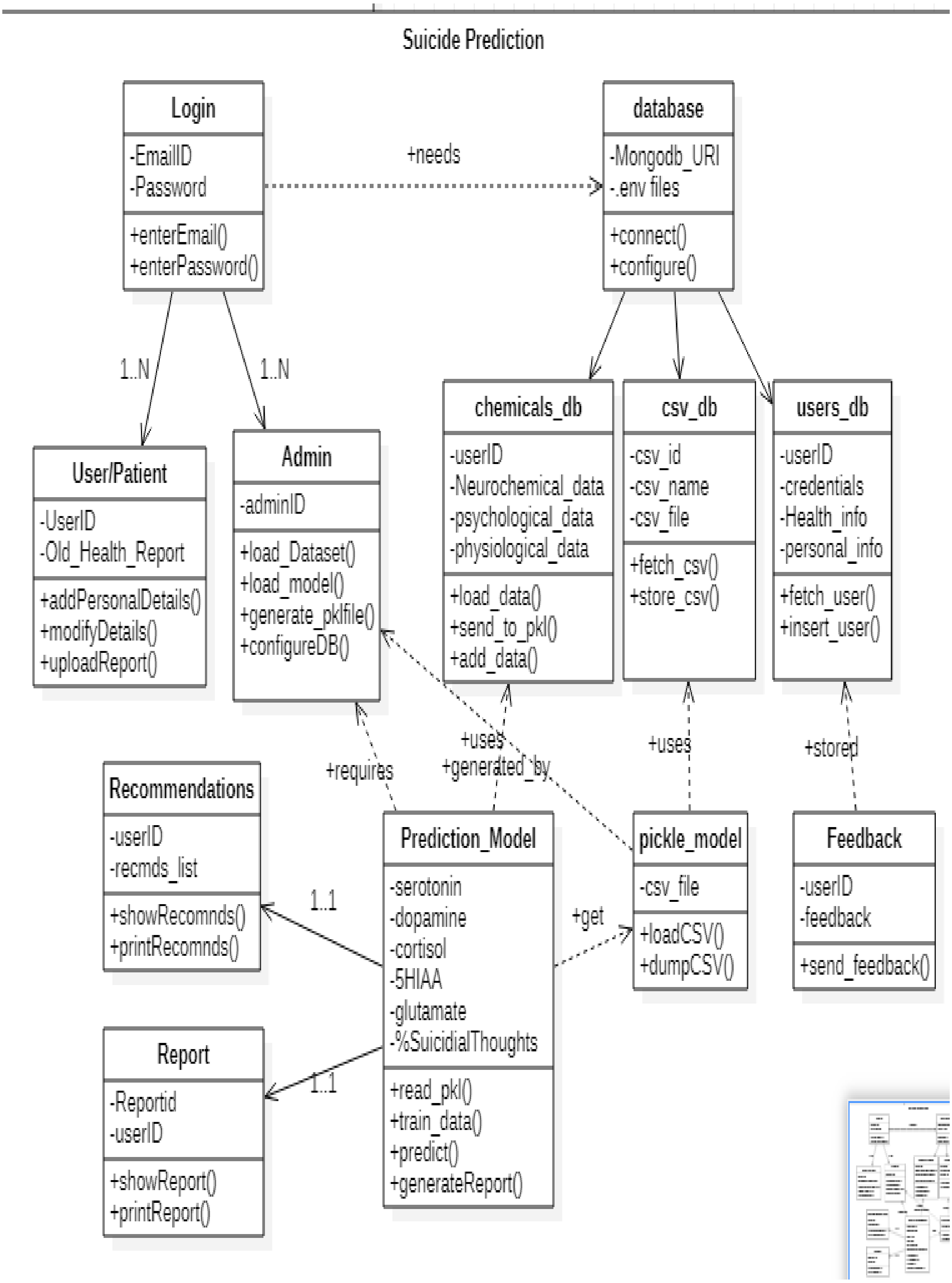
1) Activity Diagram:



2) Use Case Diagram

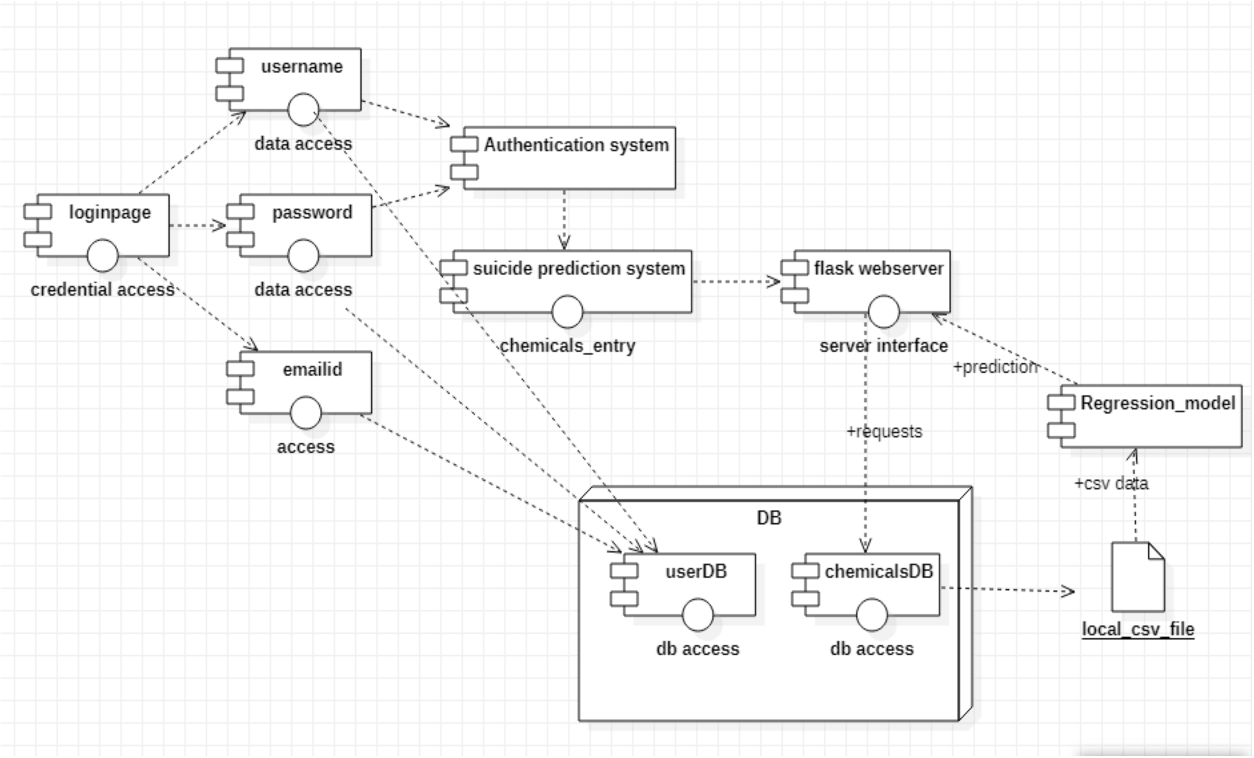


3) Class Diagram:

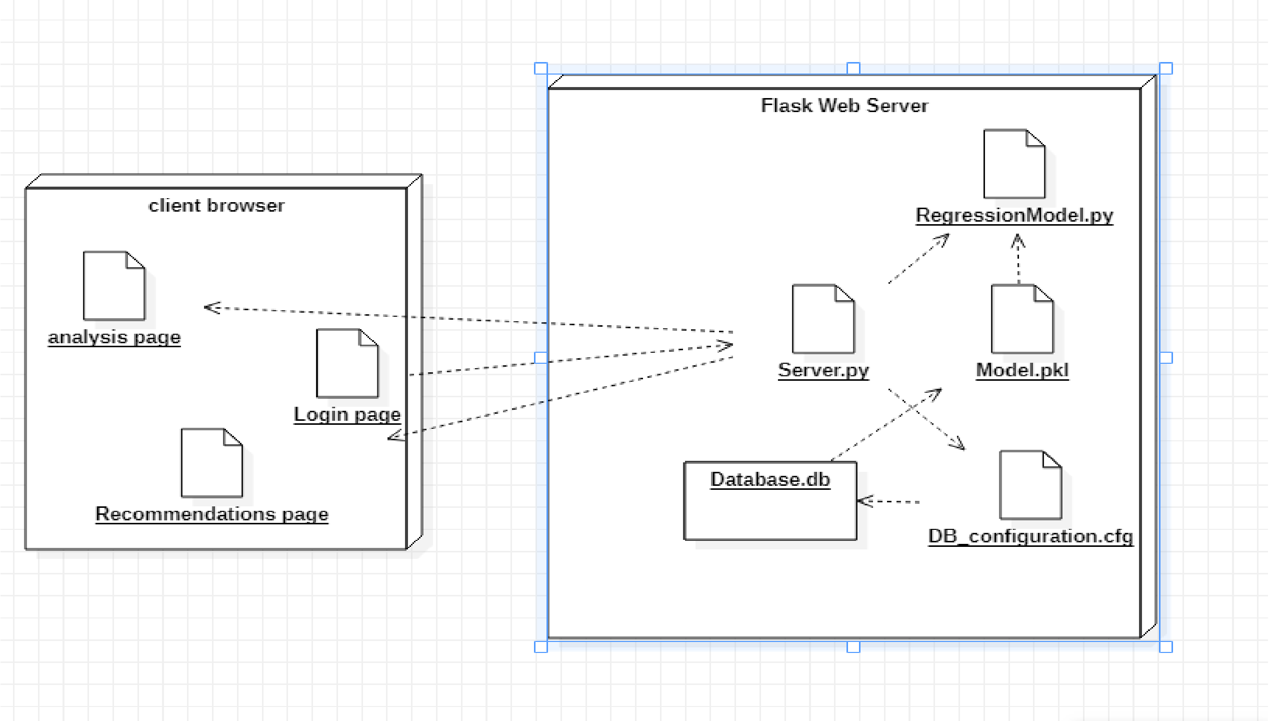


IMPLEMENTATION DIAGRAMS:

1) Component Diagram:

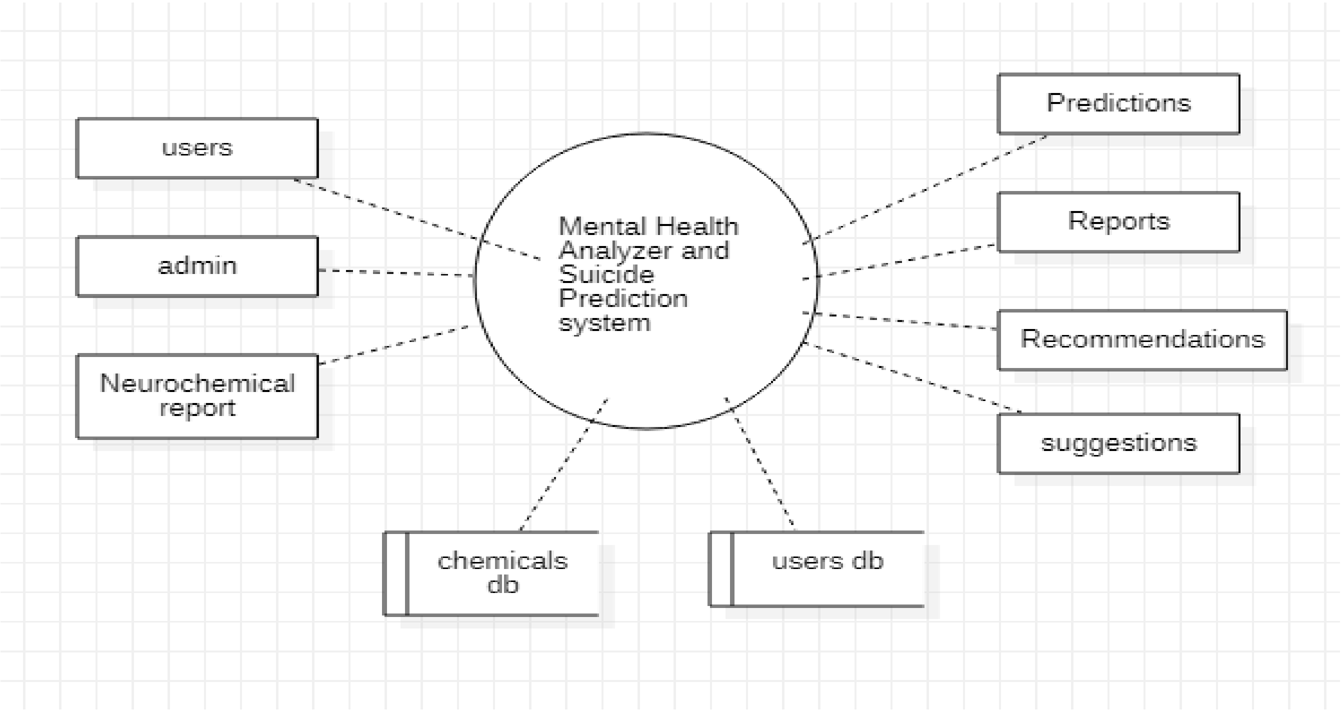


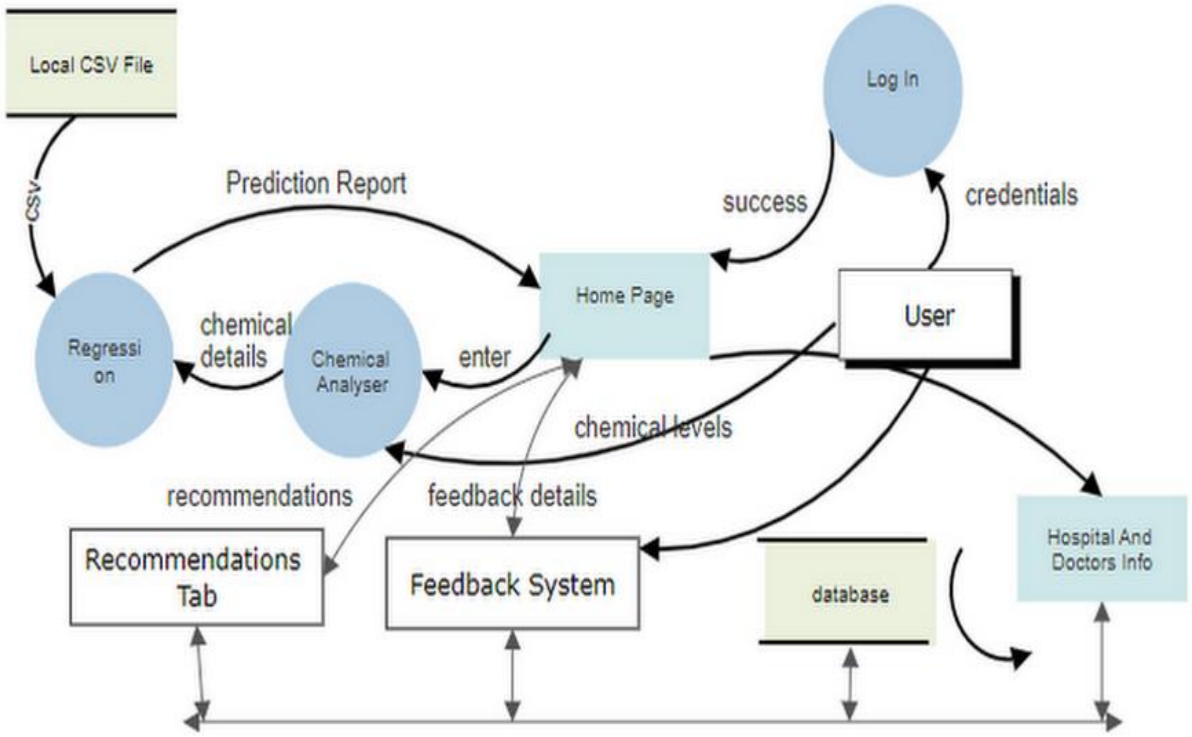
2) Deployment Diagram:



3) Data Flow Diagram:

Level 0 and Level 1 DFD respectively





DEVELOPMENT:

I. Implementation:

The Web application is implemented based on MVC and REST principles. Machine Learning libraries like sklearn, numpy and pandas are used to implement the business logic

Technology Stack Used:

Frontend : HTML5, CSS3, Materialize CSS Framework are used to implement the main interface and other GUI designs

Backend : Flask and Python are extensively used for implementing the web server and regression model respectively

Database : MongoDB for storing JSON documents

Libraries : sklearn, numpy, pandas, pymongo, mongoengine, flask

IDEs : Jupyter for model visualization & pycharm for coding

Software and Hardware Requirements:

Software : The application supports all web browsers and it smoothly runs on Google chrome and Mozilla Firefox

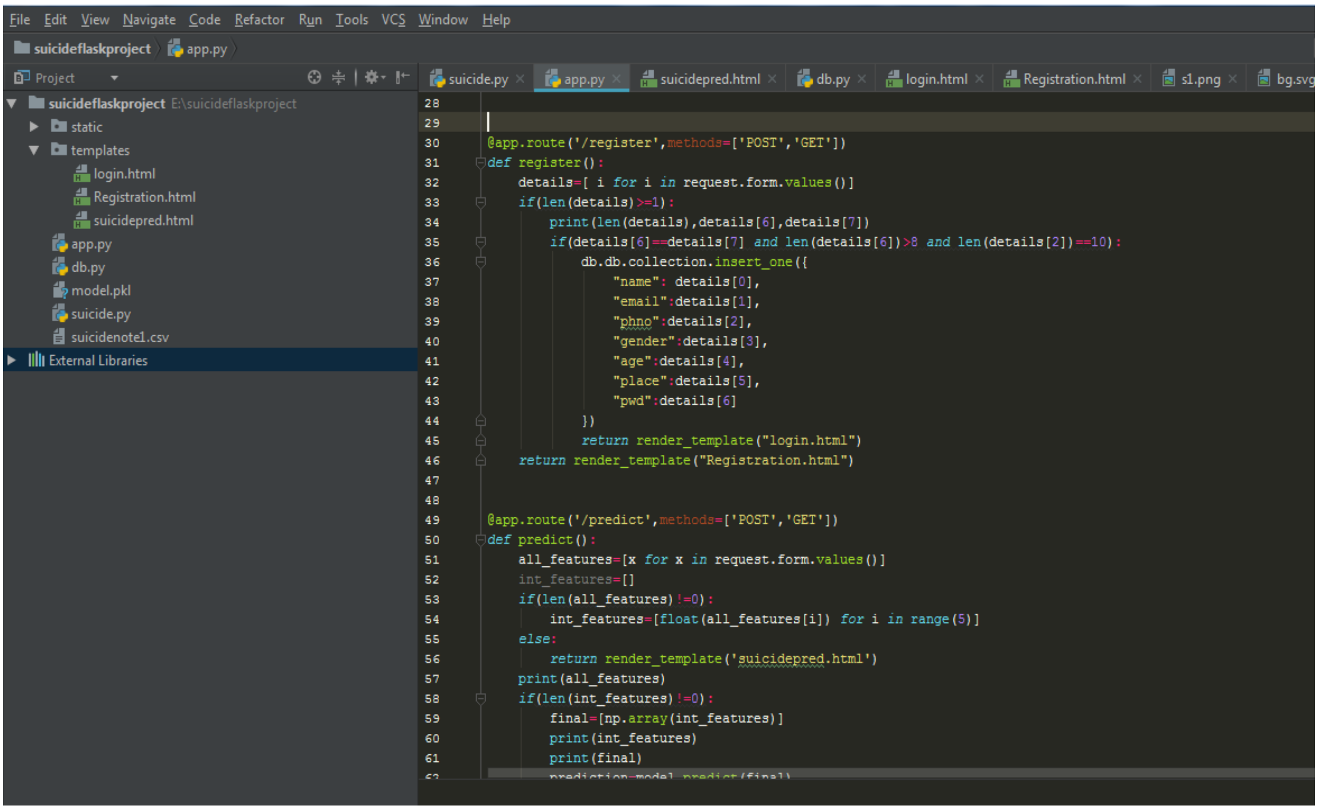
Hardware : Any Operating system that has a cookies enabled web browser installed

Acronyms used:

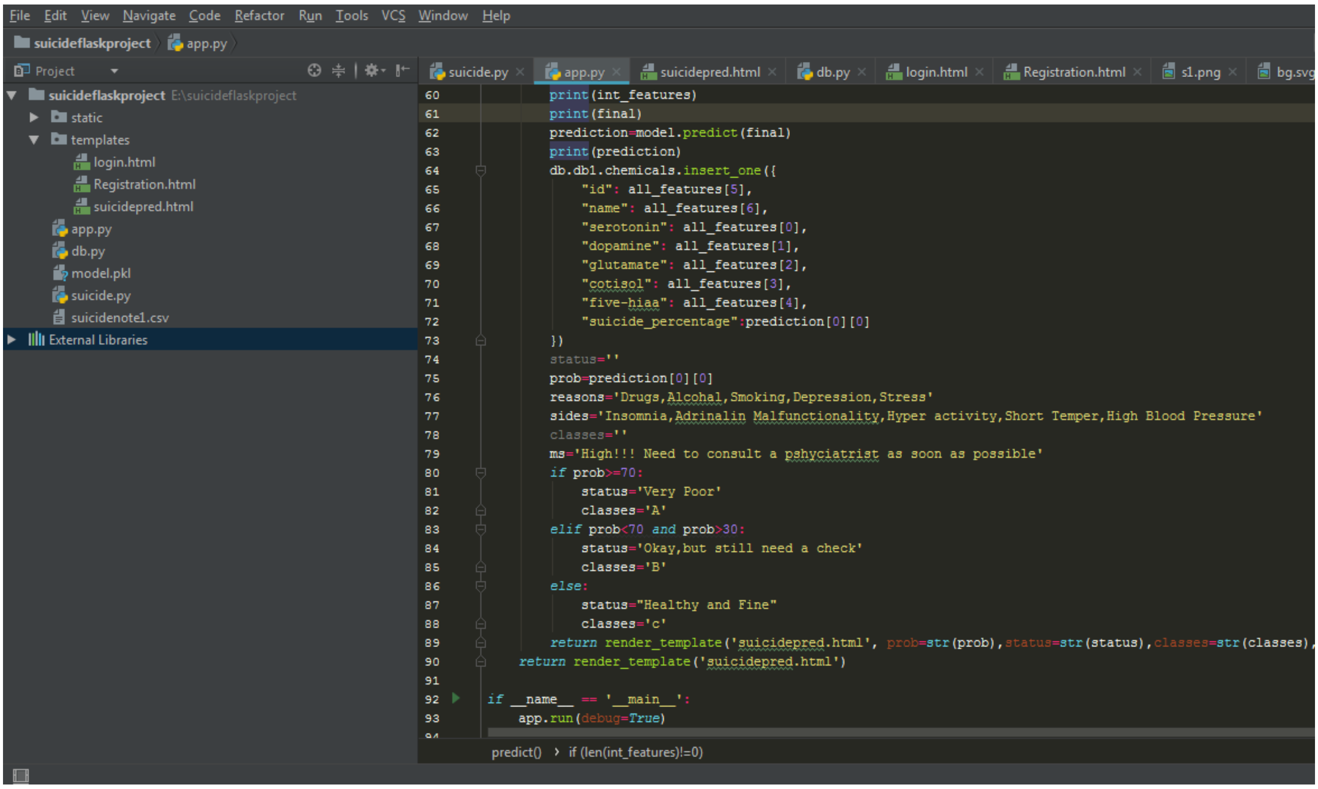
MHA	: Mental Health Analyzer
JSON	: JavaScript Object Notation
API	: Application Programming Interface
REST	: Representational State Transfer

Application Code:

app.py

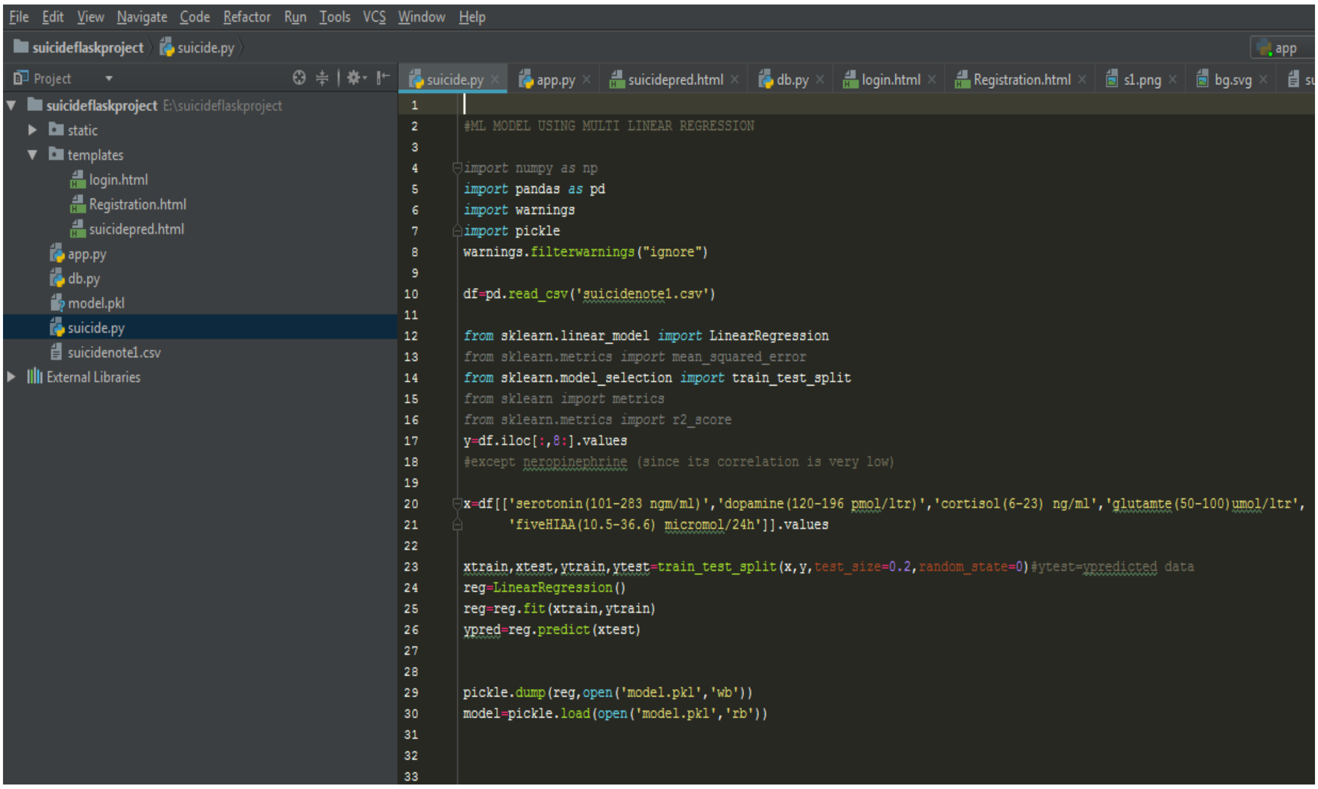


```python

@app.route('/register',methods=['POST','GET'])
def register():
    details=[ i for i in request.form.values()]
    if(len(details)>=1):
        print(len(details),details[6],details[7])
        if(details[6]==details[7] and len(details[6])>8 and len(details[2])==10):
            db.db.collection.insert_one({
                "name": details[0],
                "email":details[1],
                "phno":details[2],
                "gender":details[3],
                "age":details[4],
                "place":details[5],
                "pwd":details[6]
            })
            return render_template("login.html")
    return render_template("Registration.html")


@app.route('/predict',methods=['POST','GET'])
def predict():
    all_features=[x for x in request.form.values()]
    int_features=[]
    if(len(all_features)!=0):
        int_features=[float(all_features[i]) for i in range(5)]
    else:
        return render_template('suicidepred.html')
    print(all_features)
    if(len(int_features)!=0):
        final=[np.array(int_features)]
        print(int_features)
        print(final)
        prediction=model.predict(final)
```



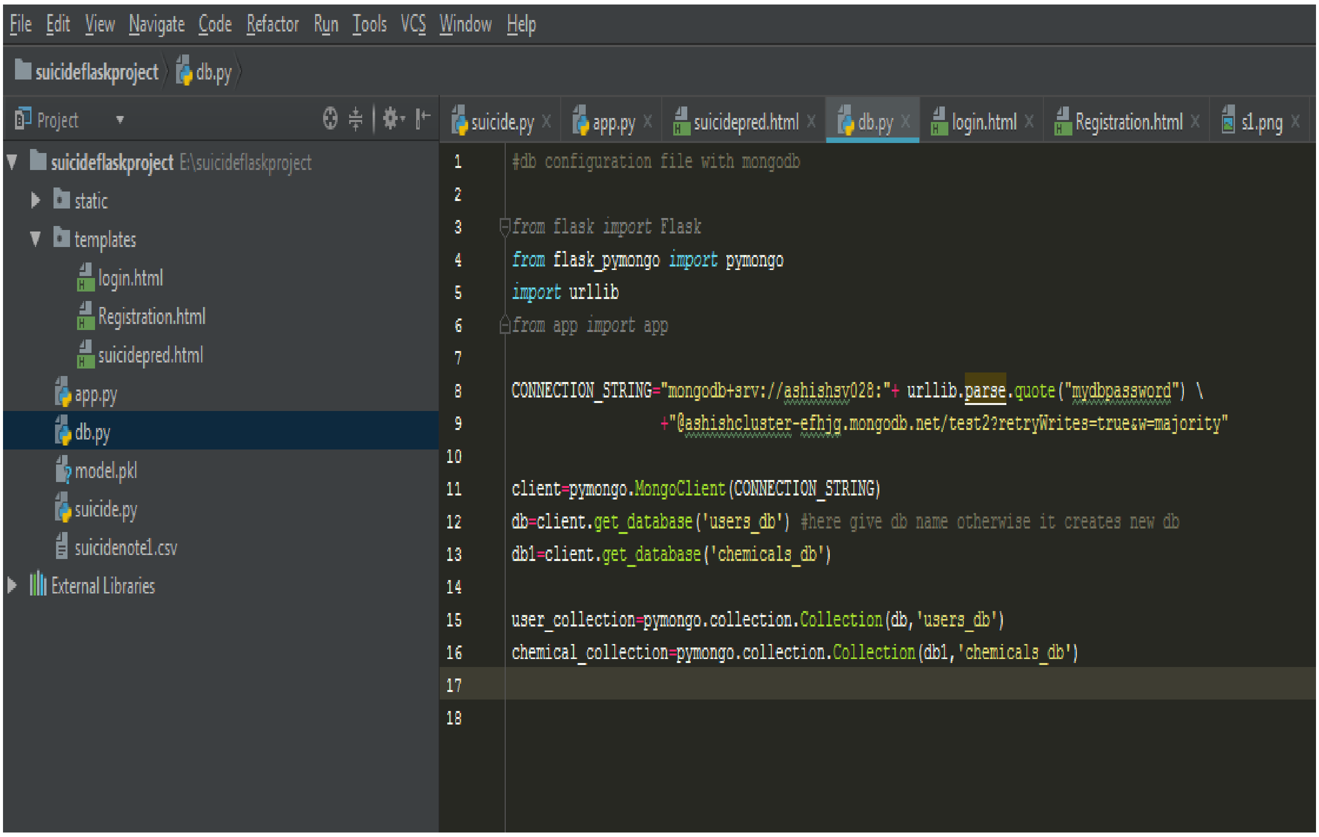
```python
        print(int_features)
        print(final)
        prediction=model.predict(final)
        print(prediction)
        db.db1.chemicals.insert_one({
            "id": all_features[5],
            "name": all_features[6],
            "serotonin": all_features[0],
            "dopamine": all_features[1],
            "glutamate": all_features[2],
            "cotisol": all_features[3],
            "five-hiaa": all_features[4],
            "suicide_percentage":prediction[0][0]
        })
        status=''
        prob=prediction[0][0]
        reasons='Drugs,Alcohal,Smoking,Depression,Stress'
        sides='Insomnia,Adrinalin Malfunctionality,Hyper activity,Short Temper,High Blood Pressure'
        classes=''
        ms='High!!! Need to consult a pshyciatrist as soon as possible'
        if prob>=70:
            status='Very Poor'
            classes='A'
        elif prob<70 and prob>30:
            status='Okay,but still need a check'
            classes='B'
        else:
            status="Healthy and Fine"
            classes='c'
        return render_template('suicidepred.html', prob=str(prob),status=str(status),classes=str(classes),
    return render_template('suicidepred.html')

if __name__ == '__main__':
    app.run(debug=True)
```

Suicide.py



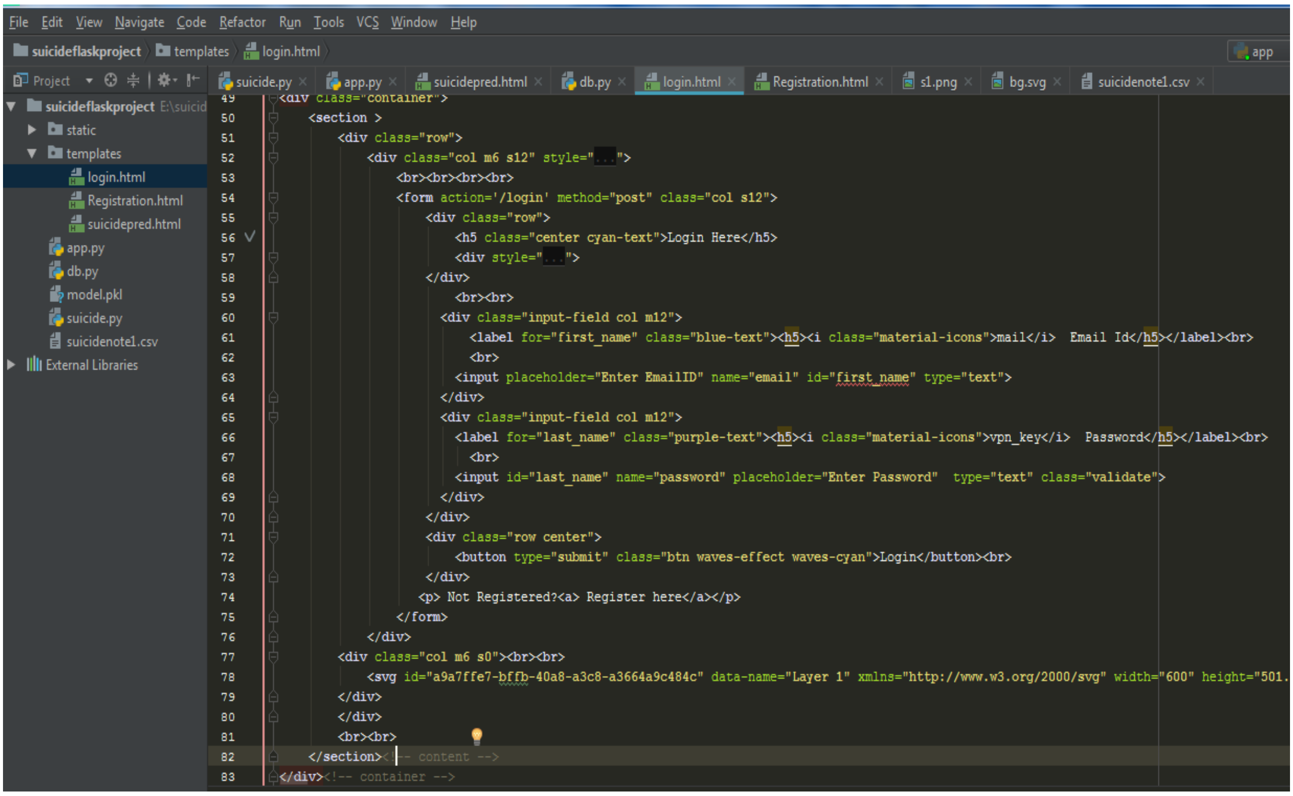
```python

#ML MODEL USING MULTI LINEAR REGRESSION

import numpy as np
import pandas as pd
import warnings
import pickle
warnings.filterwarnings("ignore")

df=pd.read_csv('suicidenote1.csv')

from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
from sklearn.model_selection import train_test_split
from sklearn import metrics
from sklearn.metrics import r2_score
y=df.iloc[:,8:].values
#except neropinephrine (since its correlation is very low)

x=df[['serotonin(101-283 ngm/ml)','dopamine(120-196 pmol/ltr)','cortisol(6-23) ng/ml','glutamte(50-100)umol/ltr',
      'fiveHIAA(10.5-36.6) micromol/24h']].values

xtrain,xtest,ytrain,ytest=train_test_split(x,y,test_size=0.2,random_state=0)#ytest=ypredicted data
reg=LinearRegression()
reg=reg.fit(xtrain,ytrain)
ypred=reg.predict(xtest)


pickle.dump(reg,open('model.pkl','wb'))
model=pickle.load(open('model.pkl','rb'))
```

Db.py



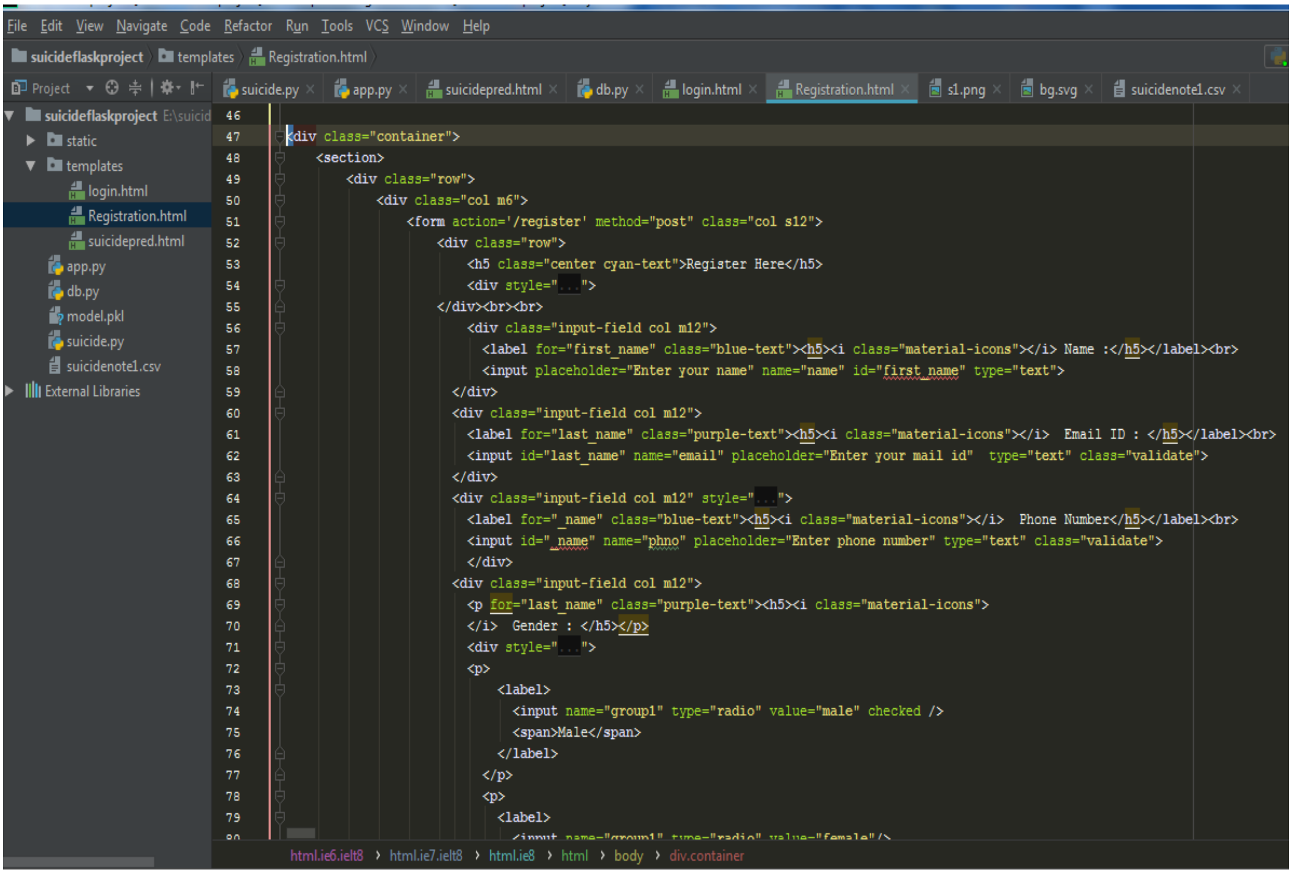
```python
#db configuration file with mongodb

from flask import Flask
from flask_pymongo import pymongo
import urllib
from app import app

CONNECTION_STRING="mongodb+srv://ashishsv028:"+ urllib.parse.quote("mydbpassword") \
                 +"@ashishcluster-efhjg.mongodb.net/test2?retryWrites=true&w=majority"

client=pymongo.MongoClient(CONNECTION_STRING)
db=client.get_database('users_db') #here give db name otherwise it creates new db
db1=client.get_database('chemicals_db')

user_collection=pymongo.collection.Collection(db,'users_db')
chemical_collection=pymongo.collection.Collection(db1,'chemicals_db')
```

Login.HTML



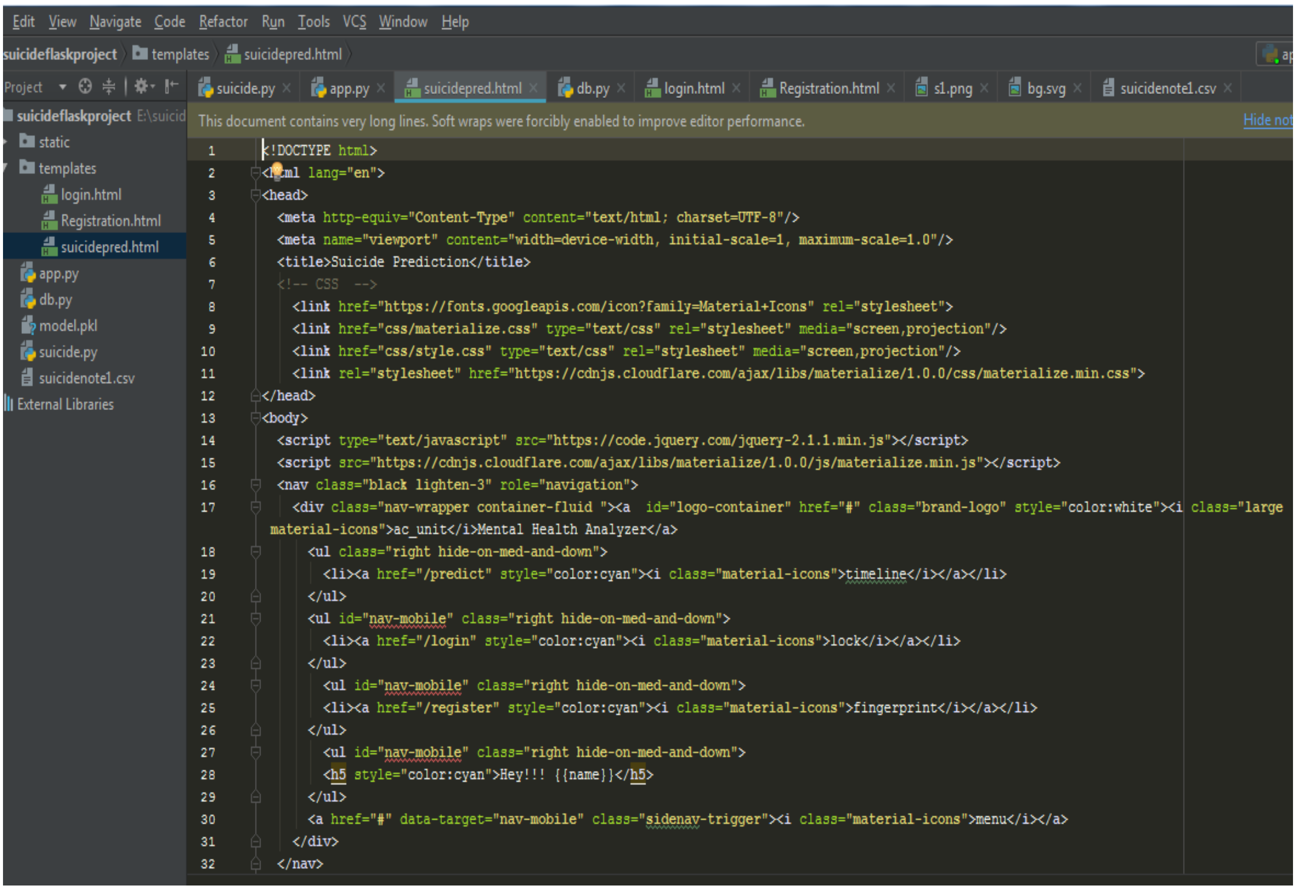
```html
<div class="container">
    <section >
        <div class="row">
            <div class="col m6 s12" style="...">
                <br><br><br><br>
                <form action='/login' method="post" class="col s12">
                    <div class="row">
                        <h5 class="center cyan-text">Login Here</h5>
                        <div style="...">
                    </div>
                        <br><br>
                      <div class="input-field col m12">
                          <label for="first_name" class="blue-text"><h5><i class="material-icons">mail</i>  Email Id</h5></label><br>
                          <br>
                        <input placeholder="Enter EmailID" name="email" id="first_name" type="text">
                      </div>
                      <div class="input-field col m12">
                        <label for="last_name" class="purple-text"><h5><i class="material-icons">vpn_key</i>  Password</h5></label><br>
                          <br>
                        <input id="last_name" name="password" placeholder="Enter Password"  type="text" class="validate">
                      </div>
                    </div>
                    <div class="row center">
                        <button type="submit" class="btn waves-effect waves-cyan">Login</button><br>
                    </div>
                   <p> Not Registered?<a> Register here</a></p>
                </form>
            </div>
        <div class="col m6 s0"><br><br>
            <svg id="a9a7ffe7-bffb-40a8-a3c8-a3664a9c484c" data-name="Layer 1" xmlns="http://www.w3.org/2000/svg" width="600" height="501.
        </div>
        </div>
        <br><br>
    </section><!-- content -->
</div><!-- container -->
```

Register.HTML



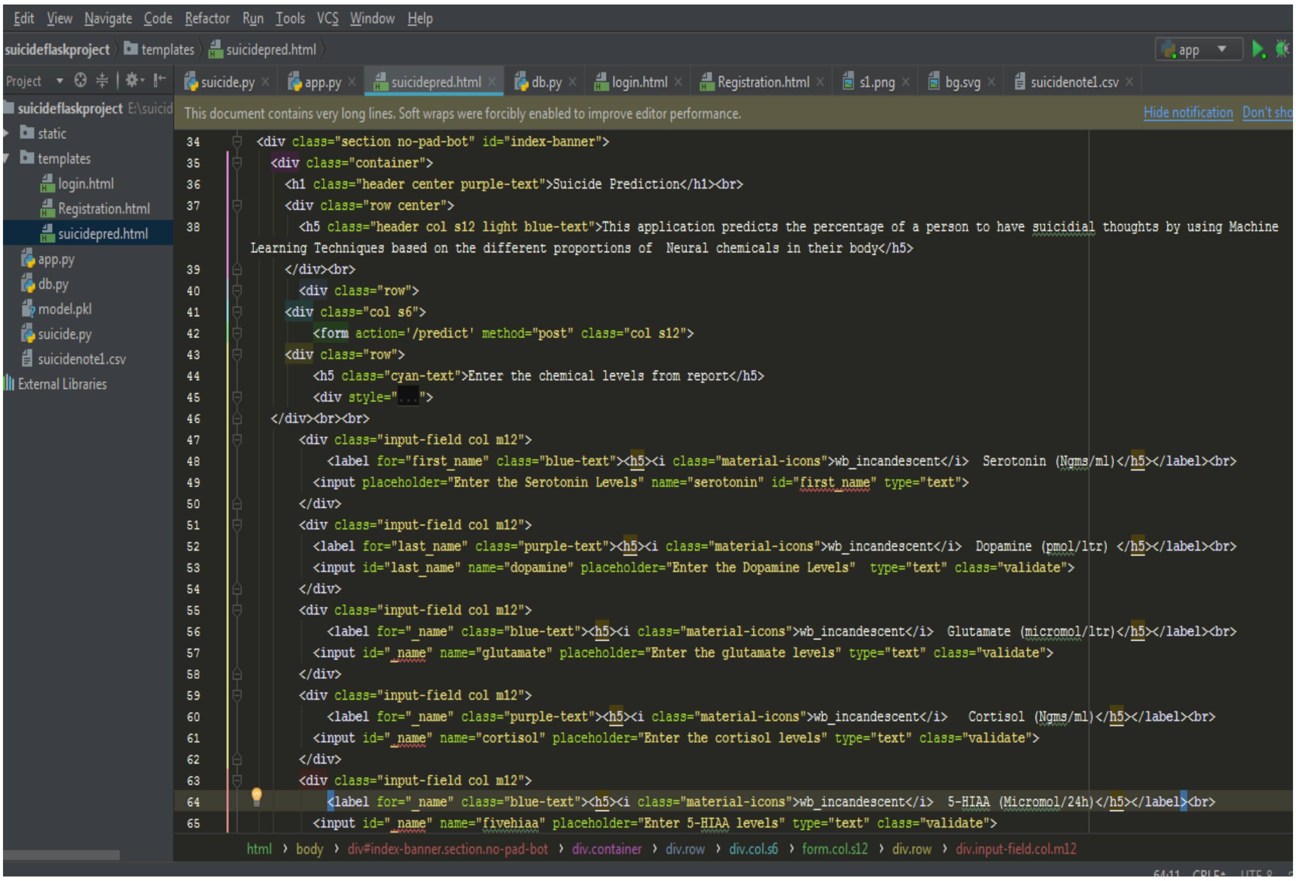
```html

<div class="container">
    <section>
        <div class="row">
            <div class="col m6">
                <form action='/register' method="post" class="col s12">
                    <div class="row">
                        <h5 class="center cyan-text">Register Here</h5>
                        <div style="...">
                    </div><br><br>
                        <div class="input-field col m12">
                          <label for="first_name" class="blue-text"><h5><i class="material-icons"></i> Name :</h5></label><br>
                          <input placeholder="Enter your name" name="name" id="first_name" type="text">
                      </div>
                      <div class="input-field col m12">
                        <label for="last_name" class="purple-text"><h5><i class="material-icons"></i>  Email ID : </h5></label><br>
                        <input id="last_name" name="email" placeholder="Enter your mail id"  type="text" class="validate">
                      </div>
                      <div class="input-field col m12" style="...">
                        <label for="_name" class="blue-text"><h5><i class="material-icons"></i>  Phone Number</h5></label><br>
                        <input id="_name" name="phno" placeholder="Enter phone number" type="text" class="validate">
                        </div>
                      <div class="input-field col m12">
                        <p for="last_name" class="purple-text"><h5><i class="material-icons">
                        </i>  Gender : </h5></p>
                        <div style="...">
                        <p>
                            <label>
                              <input name="group1" type="radio" value="male" checked />
                              <span>Male</span>
                            </label>
                          </p>
                          <p>
                            <label>
                              <input name="group1" type="radio" value="female"/>
```

prediction.html


Edit View Navigate Code Refactor Run Tools VCS Window Help
suicideflaskproject templates suicidepred.html
Project
suicide.py app.py suicidepred.html db.py login.html Registration.html s1.png bg.svg suicidenote1.csv
suicideflaskproject E:\suicid
static
templates
login.html
Registration.html
suicidepred.html
app.py
db.py
model.pkl
suicide.py
suicidenote1.csv
External Libraries
Hide not

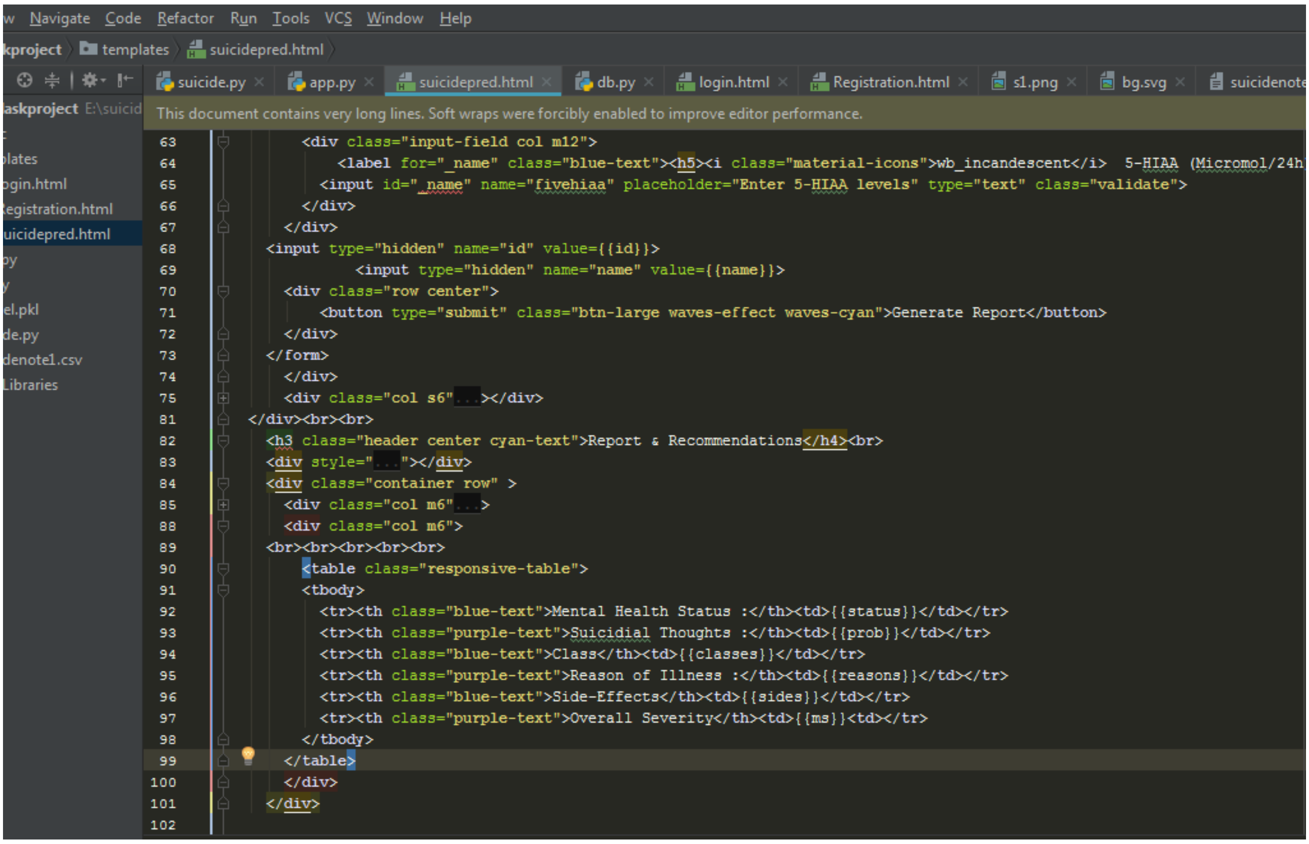
This document contains very long lines. Soft wraps were forcibly enabled to improve editor performance.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
  <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1.0"/>
  <title>Suicide Prediction</title>
  <!-- CSS  -->
    <link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet">
    <link href="css/materialize.css" type="text/css" rel="stylesheet" media="screen,projection"/>
    <link href="css/style.css" type="text/css" rel="stylesheet" media="screen,projection"/>
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/css/materialize.min.css">
</head>
<body>
  <script type="text/javascript" src="https://code.jquery.com/jquery-2.1.1.min.js"></script>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/js/materialize.min.js"></script>
  <nav class="black lighten-3" role="navigation">
    <div class="nav-wrapper container-fluid "><a  id="logo-container" href="#" class="brand-logo" style="color:white"><i class="large material-icons">ac_unit</i>Mental Health Analyzer</a>
      <ul class="right hide-on-med-and-down">
        <li><a href="/predict" style="color:cyan"><i class="material-icons">timeline</i></a></li>
      </ul>
      <ul id="nav-mobile" class="right hide-on-med-and-down">
        <li><a href="/login" style="color:cyan"><i class="material-icons">lock</i></a></li>
      </ul>
        <ul id="nav-mobile" class="right hide-on-med-and-down">
        <li><a href="/register" style="color:cyan"><i class="material-icons">fingerprint</i></a></li>
      </ul>
        <ul id="nav-mobile" class="right hide-on-med-and-down">
        <h5 style="color:cyan">Hey!!! {{name}}</h5>
      </ul>
      <a href="#" data-target="nav-mobile" class="sidenav-trigger"><i class="material-icons">menu</i></a>
    </div>
  </nav>
```


Edit View Navigate Code Refactor Run Tools VCS Window Help
suicideflaskproject templates suicidepred.html
app
Project
suicide.py app.py suicidepred.html db.py login.html Registration.html s1.png bg.svg suicidenote1.csv
suicideflaskproject E:\suicid
static
templates
login.html
Registration.html
suicidepred.html
app.py
db.py
model.pkl
suicide.py
suicidenote1.csv
External Libraries
Hide notification Don't sho
html › body › div#index-banner.section.no-pad-bot › div.container › div.row › div.col.s6 › form.col.s12 › div.row › div.input-field.col.m12

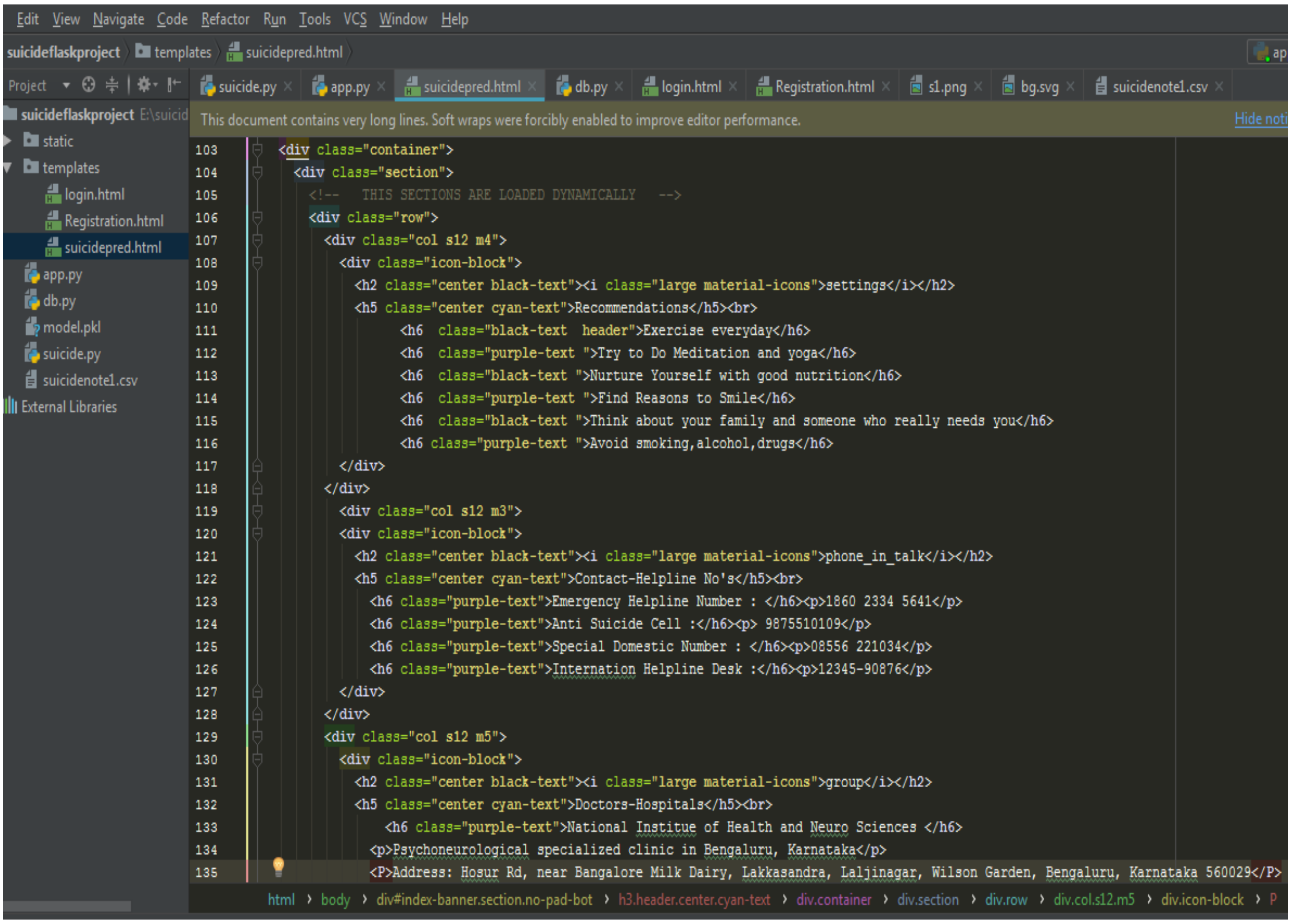
This document contains very long lines. Soft wraps were forcibly enabled to improve editor performance.

```
  <div class="section no-pad-bot" id="index-banner">
    <div class="container">
      <h1 class="header center purple-text">Suicide Prediction</h1><br>
      <div class="row center">
        <h5 class="header col s12 light blue-text">This application predicts the percentage of a person to have suicidial thoughts by using Machine Learning Techniques based on the different proportions of  Neural chemicals in their body</h5>
      </div><br>
        <div class="row">
      <div class="col s6">
          <form action='/predict' method="post" class="col s12">
      <div class="row">
          <h5 class="cyan-text">Enter the chemical levels from report</h5>
          <div style="...">
    </div><br><br>
        <div class="input-field col m12">
            <label for="first_name" class="blue-text"><h5><i class="material-icons">wb_incandescent</i>  Serotonin (Ngms/ml)</h5></label><br>
          <input placeholder="Enter the Serotonin Levels" name="serotonin" id="first_name" type="text">
        </div>
        <div class="input-field col m12">
          <label for="last_name" class="purple-text"><h5><i class="material-icons">wb_incandescent</i>  Dopamine (pmol/ltr) </h5></label><br>
          <input id="last_name" name="dopamine" placeholder="Enter the Dopamine Levels"  type="text" class="validate">
        </div>
        <div class="input-field col m12">
            <label for="_name" class="blue-text"><h5><i class="material-icons">wb_incandescent</i>  Glutamate (micromol/ltr)</h5></label><br>
          <input id="_name" name="glutamate" placeholder="Enter the glutamate levels" type="text" class="validate">
        </div>
        <div class="input-field col m12">
            <label for="_name" class="purple-text"><h5><i class="material-icons">wb_incandescent</i>   Cortisol (Ngms/ml)</h5></label><br>
          <input id="_name" name="cortisol" placeholder="Enter the cortisol levels" type="text" class="validate">
        </div>
        <div class="input-field col m12">
            <label for="_name" class="blue-text"><h5><i class="material-icons">wb_incandescent</i>  5-HIAA (Micromol/24h)</h5></label><br>
          <input id="_name" name="fivehiaa" placeholder="Enter 5-HIAA levels" type="text" class="validate">
```


w Navigate Code Refactor Run Tools VCS Window Help
kproject templates suicidepred.html
suicide.py app.py suicidepred.html db.py login.html Registration.html s1.png bg.svg suicidenote
askproject E:\suicid
plates
ogin.html
Registration.html
uicidepred.html
py
y
el.pkl
de.py
denote1.csv
Libraries

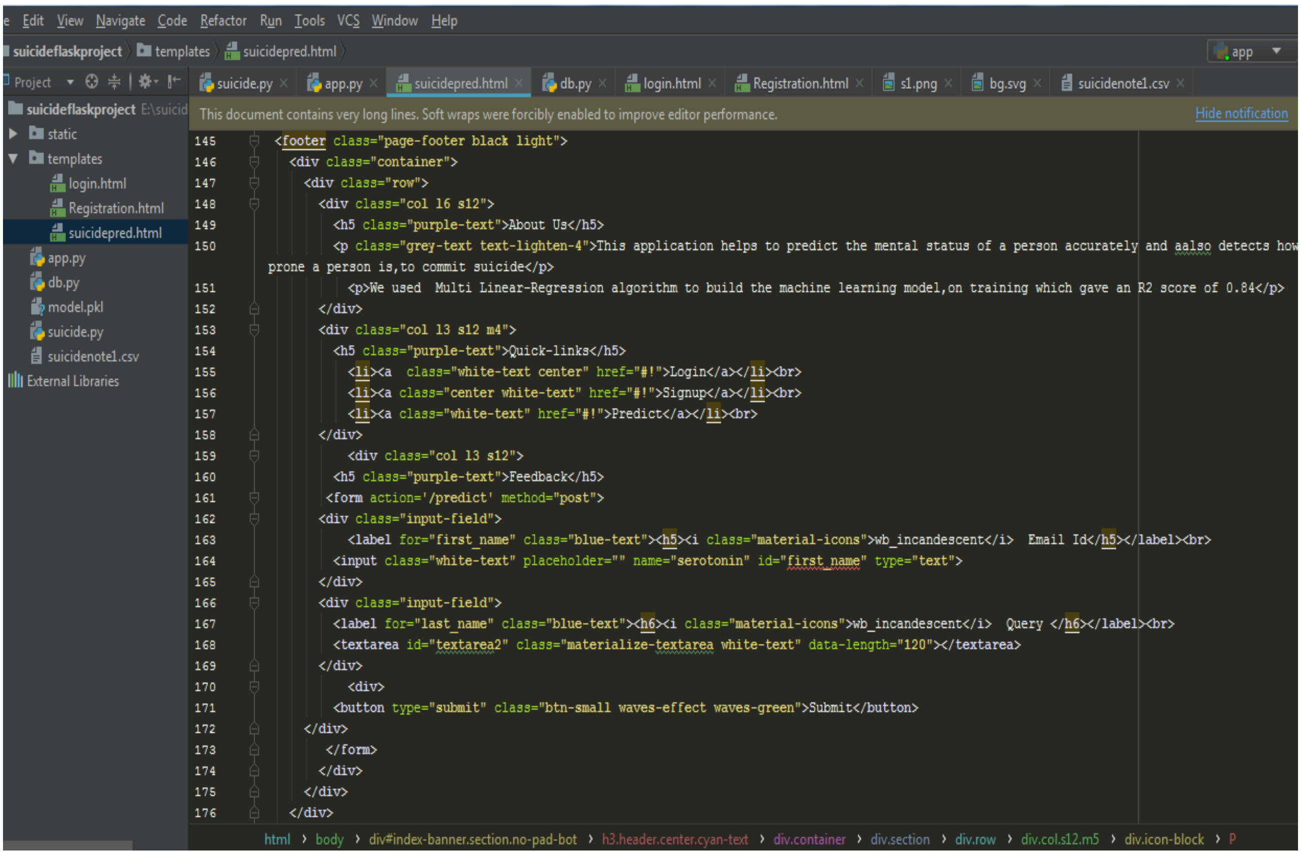
This document contains very long lines. Soft wraps were forcibly enabled to improve editor performance.

```
        <div class="input-field col m12">
            <label for="_name" class="blue-text"><h5><i class="material-icons">wb_incandescent</i>  5-HIAA (Micromol/24h
          <input id="_name" name="fivehiaa" placeholder="Enter 5-HIAA levels" type="text" class="validate">
        </div>
      </div>
    <input type="hidden" name="id" value={{id}}>
             <input type="hidden" name="name" value={{name}}>
      <div class="row center">
          <button type="submit" class="btn-large waves-effect waves-cyan">Generate Report</button>
      </div>
    </form>
      </div>
      <div class="col s6"...></div>
  </div><br><br>
    <h3 class="header center cyan-text">Report & Recommendations</h4><br>
    <div style="..."></div>
    <div class="container row" >
      <div class="col m6"...>
      <div class="col m6">
    <br><br><br><br><br>
        <table class="responsive-table">
        <tbody>
          <tr><th class="blue-text">Mental Health Status :</th><td>{{status}}</td></tr>
          <tr><th class="purple-text">Suicidial Thoughts :</th><td>{{prob}}</td></tr>
          <tr><th class="blue-text">Class</th><td>{{classes}}</td></tr>
          <tr><th class="purple-text">Reason of Illness :</th><td>{{reasons}}</td></tr>
          <tr><th class="blue-text">Side-Effects</th><td>{{sides}}</td></tr>
          <tr><th class="purple-text">Overall Severity</th><td>{{ms}}<td></tr>
        </tbody>
      </table>
      </div>
    </div>
```


Edit View Navigate Code Refactor Run Tools VCS Window Help
suicideflaskproject templates suicidepred.html
Project
suicide.py app.py suicidepred.html db.py login.html Registration.html s1.png bg.svg suicidenote1.csv
suicideflaskproject E:\suicid
static
templates
login.html
Registration.html
suicidepred.html
app.py
db.py
model.pkl
suicide.py
suicidenote1.csv
External Libraries
Hide noti
html › body › div#index-banner.section.no-pad-bot › h3.header.center.cyan-text › div.container › div.section › div.row › div.col.s12.m5 › div.icon-block › P

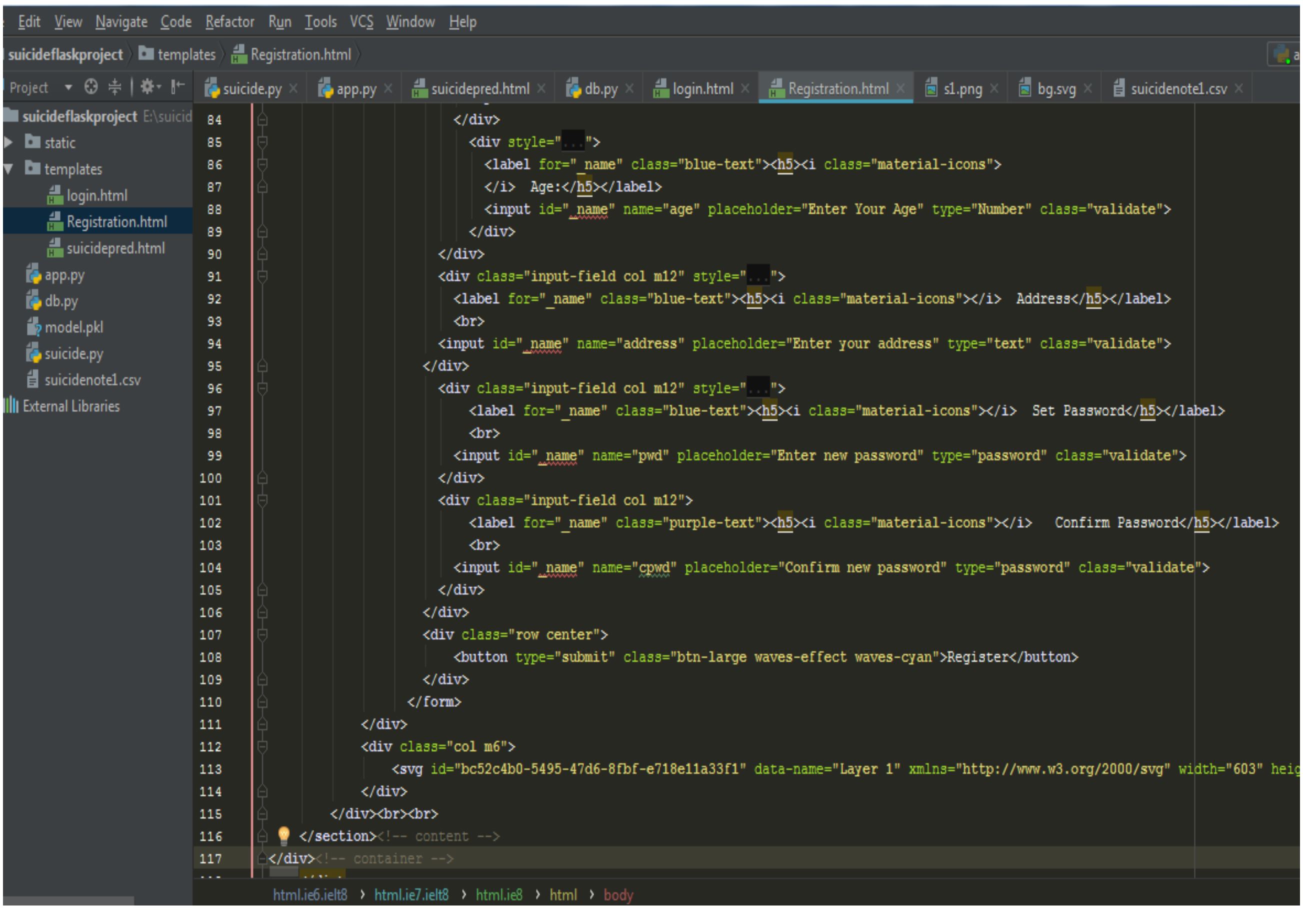
This document contains very long lines. Soft wraps were forcibly enabled to improve editor performance.

```
<div class="container">
  <div class="section">
    <!--   THIS SECTIONS ARE LOADED DYNAMICALLY   -->
    <div class="row">
      <div class="col s12 m4">
        <div class="icon-block">
          <h2 class="center black-text"><i class="large material-icons">settings</i></h2>
          <h5 class="center cyan-text">Recommendations</h5><br>
                <h6  class="black-text  header">Exercise everyday</h6>
                <h6  class="purple-text ">Try to Do Meditation and yoga</h6>
                <h6  class="black-text ">Nurture Yourself with good nutrition</h6>
                <h6  class="purple-text ">Find Reasons to Smile</h6>
                <h6  class="black-text ">Think about your family and someone who really needs you</h6>
                <h6 class="purple-text ">Avoid smoking,alcohol,drugs</h6>
        </div>
      </div>
        <div class="col s12 m3">
        <div class="icon-block">
          <h2 class="center black-text"><i class="large material-icons">phone_in_talk</i></h2>
          <h5 class="center cyan-text">Contact-Helpline No's</h5><br>
            <h6 class="purple-text">Emergency Helpline Number : </h6><p>1860 2334 5641</p>
            <h6 class="purple-text">Anti Suicide Cell :</h6><p> 9875510109</p>
            <h6 class="purple-text">Special Domestic Number : </h6><p>08556 221034</p>
            <h6 class="purple-text">Internation Helpline Desk :</h6><p>12345-90876</p>
        </div>
      </div>
      <div class="col s12 m5">
        <div class="icon-block">
          <h2 class="center black-text"><i class="large material-icons">group</i></h2>
          <h5 class="center cyan-text">Doctors-Hospitals</h5><br>
              <h6 class="purple-text">National Institue of Health and Neuro Sciences </h6>
            <p>Psychoneurological specialized clinic in Bengaluru, Karnataka</p>
            <P>Address: Hosur Rd, near Bangalore Milk Dairy, Lakkasandra, Laljinagar, Wilson Garden, Bengaluru, Karnataka 560029</P>
```


e Edit View Navigate Code Refactor Run Tools VCS Window Help
suicideflaskproject templates suicidepred.html
app
Project
suicide.py app.py suicidepred.html db.py login.html Registration.html s1.png bg.svg suicidenote1.csv
suicideflaskproject E:\suicid
static
templates
login.html
Registration.html
suicidepred.html
app.py
db.py
model.pkl
suicide.py
suicidenote1.csv
External Libraries
Hide notification
html › body › div#index-banner.section.no-pad-bot › h3.header.center.cyan-text › div.container › div.section › div.row › div.col.s12.m5 › div.icon-block › P

This document contains very long lines. Soft wraps were forcibly enabled to improve editor performance.

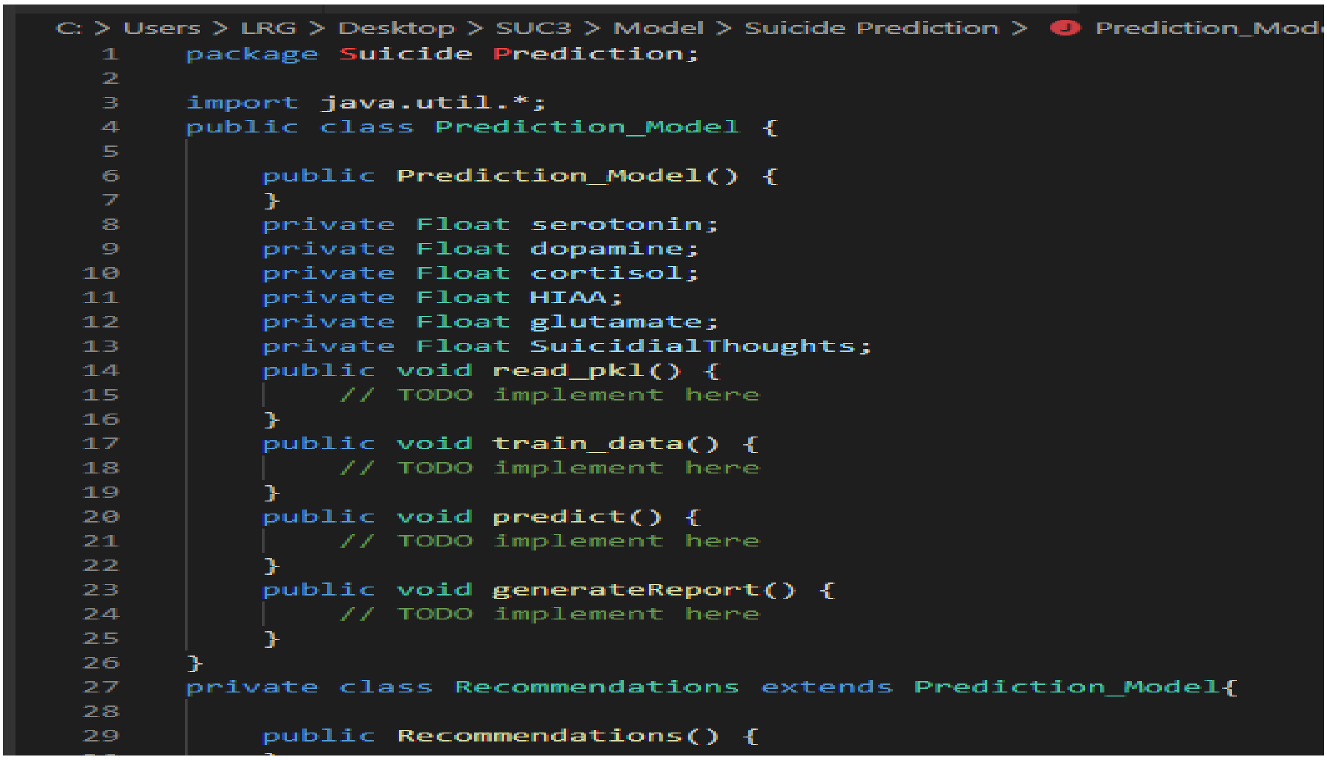
```
<footer class="page-footer black light">
  <div class="container">
    <div class="row">
      <div class="col 16 s12">
        <h5 class="purple-text">About Us</h5>
        <p class="grey-text text-lighten-4">This application helps to predict the mental status of a person accurately and aalso detects how
prone a person is,to commit suicide</p>
          <p>We used  Multi Linear-Regression algorithm to build the machine learning model,on training which gave an R2 score of 0.84</p>
      </div>
      <div class="col 13 s12 m4">
        <h5 class="purple-text">Quick-links</h5>
          <li><a  class="white-text center" href="#!">Login</a></li><br>
          <li><a class="center white-text" href="#!">Signup</a></li><br>
          <li><a class="white-text" href="#!">Predict</a></li><br>
      </div>
          <div class="col 13 s12">
        <h5 class="purple-text">Feedback</h5>
       <form action='/predict' method="post">
      <div class="input-field">
          <label for="first_name" class="blue-text"><h5><i class="material-icons">wb_incandescent</i>  Email Id</h5></label><br>
        <input class="white-text" placeholder="" name="serotonin" id="first_name" type="text">
      </div>
      <div class="input-field">
        <label for="last_name" class="blue-text"><h6><i class="material-icons">wb_incandescent</i>  Query </h6></label><br>
        <textarea id="textarea2" class="materialize-textarea white-text" data-length="120"></textarea>
      </div>
          <div>
        <button type="submit" class="btn-small waves-effect waves-green">Submit</button>
    </div>
       </form>
      </div>
    </div>
  </div>
```


Edit View Navigate Code Refactor Run Tools VCS Window Help
suicideflaskproject templates Registration.html
Project
suicide.py app.py suicidepred.html db.py login.html Registration.html s1.png bg.svg suicidenote1.csv
suicideflaskproject E:\suicid
static
templates
login.html
Registration.html
suicidepred.html
app.py
db.py
model.pkl
suicide.py
suicidenote1.csv
External Libraries
html.ie6.ielt8 › html.ie7.ielt8 › html.ie8 › html › body

```
                </div>
                  <div style="...">
                    <label for="_name" class="blue-text"><h5><i class="material-icons">
                    </i>  Age:</h5></label>
                    <input id="_name" name="age" placeholder="Enter Your Age" type="Number" class="validate">
                  </div>
                </div>
                <div class="input-field col m12" style="...">
                  <label for="_name" class="blue-text"><h5><i class="material-icons"></i>  Address</h5></label>
                  <br>
                <input id="_name" name="address" placeholder="Enter your address" type="text" class="validate">
              </div>
                <div class="input-field col m12" style="...">
                    <label for="_name" class="blue-text"><h5><i class="material-icons"></i>  Set Password</h5></label>
                    <br>
                  <input id="_name" name="pwd" placeholder="Enter new password" type="password" class="validate">
                </div>
                <div class="input-field col m12">
                    <label for="_name" class="purple-text"><h5><i class="material-icons"></i>   Confirm Password</h5></label>
                    <br>
                  <input id="_name" name="cpwd" placeholder="Confirm new password" type="password" class="validate">
                </div>
              </div>
              <div class="row center">
                  <button type="submit" class="btn-large waves-effect waves-cyan">Register</button>
              </div>
            </form>
          </div>
          <div class="col m6">
              <svg id="bc52c4b0-5495-47d6-8fbf-e718e11a33f1" data-name="Layer 1" xmlns="http://www.w3.org/2000/svg" width="603" heig
          </div>
        </div><br><br>
  </section><!-- content -->
</div><!-- container -->
```

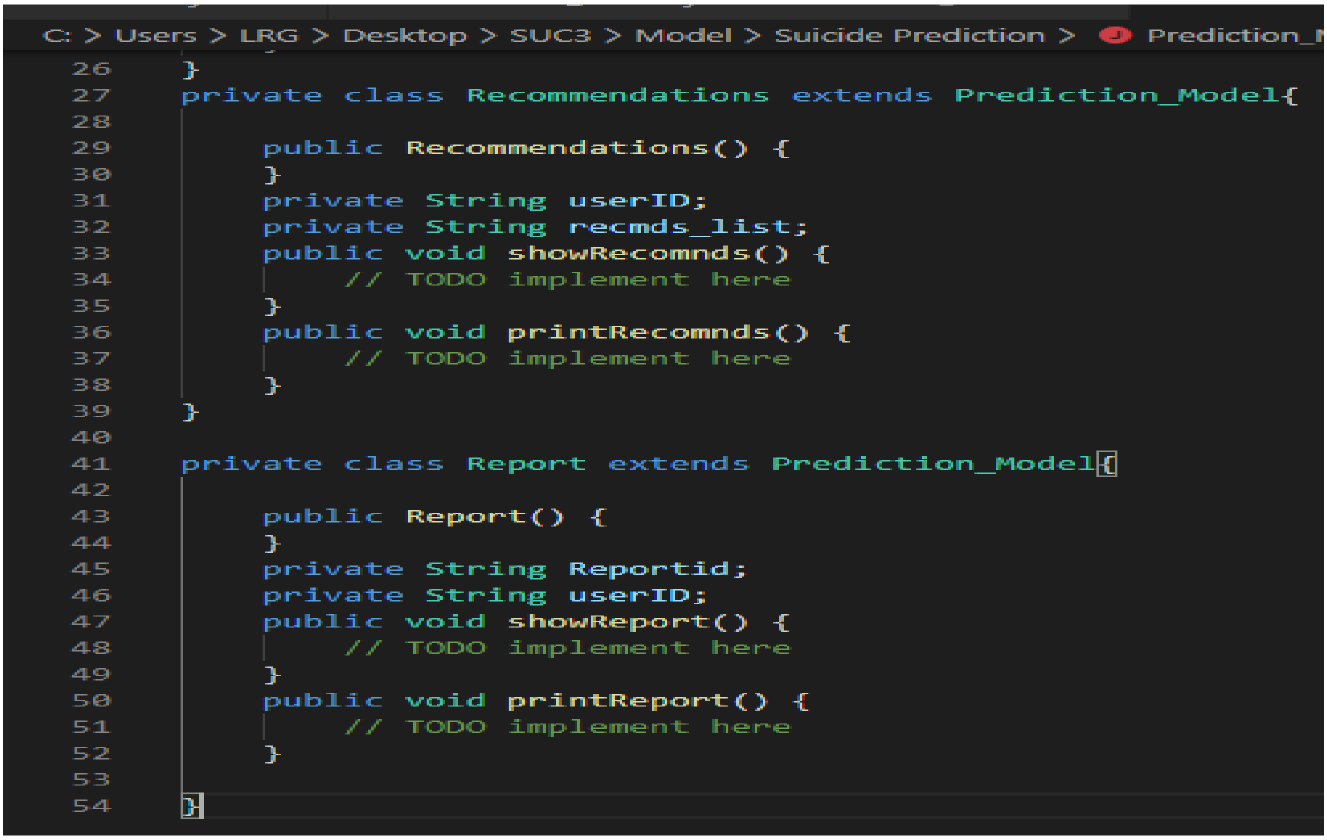
II. FORWARD ENGINEERING:

Java code generated



C: > Users > LRG > Desktop > SUC3 > Model > Suicide Prediction > Prediction_Mod

```java
package Suicide Prediction;

import java.util.*;
public class Prediction_Model {

    public Prediction_Model() {
    }
    private Float serotonin;
    private Float dopamine;
    private Float cortisol;
    private Float HIAA;
    private Float glutamate;
    private Float SuicidialThoughts;
    public void read_pkl() {
        // TODO implement here
    }
    public void train_data() {
        // TODO implement here
    }
    public void predict() {
        // TODO implement here
    }
    public void generateReport() {
        // TODO implement here
    }
}
private class Recommendations extends Prediction_Model{

    public Recommendations() {
```

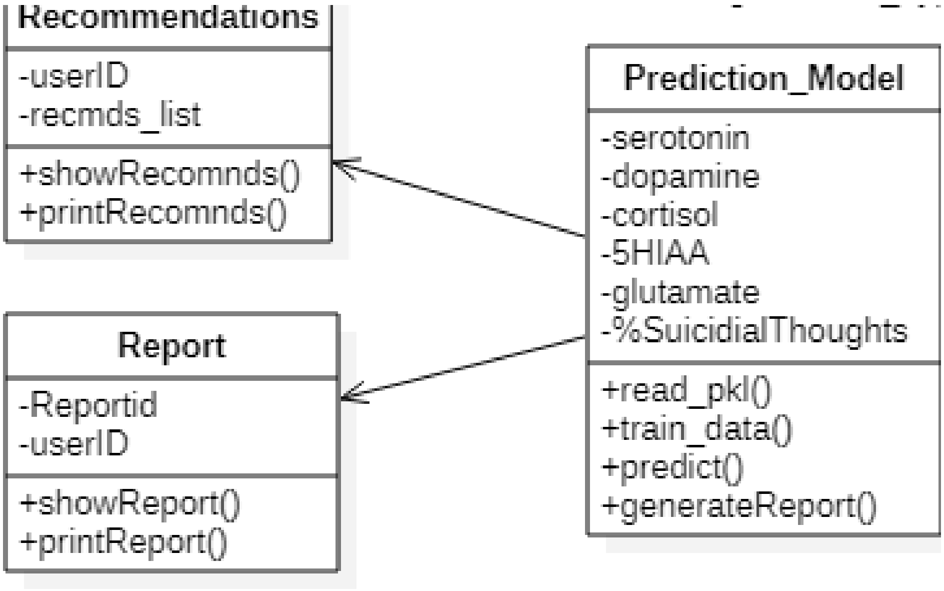


C: > Users > LRG > Desktop > SUC3 > Model > Suicide Prediction > Prediction_

```java
}
private class Recommendations extends Prediction_Model{

    public Recommendations() {
    }
    private String userID;
    private String recmds_list;
    public void showRecomnds() {
        // TODO implement here
    }
    public void printRecomnds() {
        // TODO implement here
    }
}

private class Report extends Prediction_Model{

    public Report() {
    }
    private String Reportid;
    private String userID;
    public void showReport() {
        // TODO implement here
    }
    public void printReport() {
        // TODO implement here
    }

}
```

III. REVERSE ENGINEERING:

Class diagram generated from JAVA code



TESTING:

1) Black Box Testing

What is Black Box Testing?

Black Box testing is defined as a testing technique in which functionality of the Application Under Test (AUT) is tested without looking at the internal code structure, implementation details and knowledge of internal paths of the software.

This type of testing is based entirely on software requirements and specifications. In BlackBox Testing we just focus on inputs and output of the software system without bothering about internal knowledge of the software program

Black Box Testing
Input
Black Box
Output
ReQtest

Value conditions for Prediction Analyzer:

```
from mongoengine import Document
class analyzer(Document):
    id=StringField(max_length=60,required=True,unique=True)
    name=StringField(max_length=60,required=True,unique=True)
    serotonin=FloatField(max_value=283.00,min_value=101.00)
    dopamine = FloatField(max_value=195.80, min_value=120.00)
    glutamate = FloatField(max_value=100.00, min_value=50.00)
    cortisol = FloatField(max_value=23.00, min_value=6.00)
    fivehiaa = FloatField(max_value=36.60, min_value=10.50)
    suicidePer=FloatField(max_value=100.00, min_value=0.00)
```

Black box testing using Equivalence class testing method:

Characteristic	Invalid	Valid	Invalid
Id	<15	15-60	>60
Name	<5	5-20	>20
Serotonin	<101.00	101.00-283.00	>283.00
Dopamine	<120.00	120.00-195.80	>195.80
Glutamate	<50.00	50.00-100.00	>100.00
Cortisol	<6.00	6.00-23.00	>23.00
Fivehiaa	<10.50	10.50-36.60	>36.60
SuicidePer	<1	1.00-100.00	>100.00

Sample test cases:

- **Invalid:**

Id	:	e23P45007$qw
Name	:	Sai
Serotonin	:	85.65
Dopamine	:	100.56
Glutamate	:	21.7878
Cortisol	:	5.120
Fivehiaa	:	8.456
Suicide_per	:	0.50

- **Valid:**

Id	:	e23P45007$qwe93l45e07$qwe23P
Name	:	Ashish
Serotonin	:	185.65
Dopamine	:	150.56
Glutamate	:	177.7878
Cortisol	:	15.120
Fivehiaa	:	28.456
Suicide_per	:	30.50

- **Invalid:**

Id	:	e23P45007$qwe93l45e07$qwe23P45007$qw57$q.......
Name	:	Pramodhsairamrangasatyaviswapavan
Serotonin	:	585.65
Dopamine	:	450.56
Glutamate	:	377.7878
Cortisol	:	115.120
Fivehiaa	:	248.456
Suicide_per	:	110.50

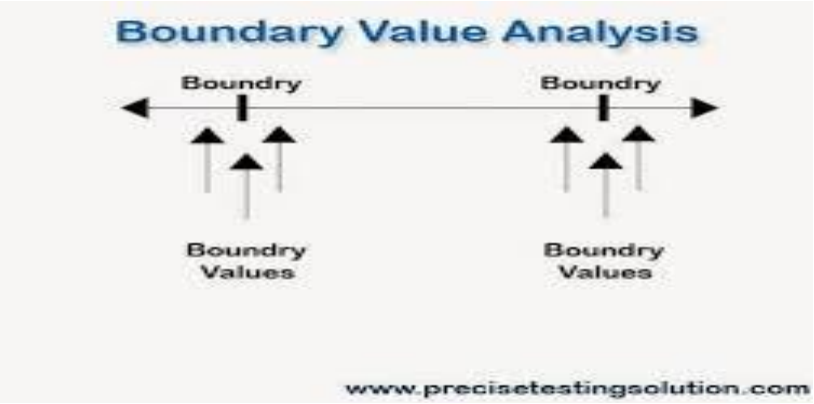
Registration Module:

```
from mongoengine import Document
class Register(Document):
  name=StringField(min_length=5,max_length=20,required=True,unique=True)
   mail=StringField(min_length=12,max_length=30,required=True,unique=True)
   phno=StringField(min_length=6,max_length=10,required=True,unique=True)
   gender=StringField(max_length=1,required=True,unique=True)
   age=IntegerField(min_val=18,max_val=80,required=True,unique=True)
   pwd=StringField(min_length=8,max_length=20,required=True,unique=True)
```

Black box testing using Boundary value analysis method:

It is widely recognized that input values at the extreme ends of the input domain cause more errors in the system. More application errors occur at the boundaries of the input domain.

Boundary Value Analysis testing technique is used to identify errors at the boundaries rather than finding those exist in the centre of the input domain. Boundary Value Analysis is the next part of equivalence portioning for designing test cases where test cases are selected at the edges of the equivalence classes.



Characteristics	Invalid (min-1)	Valid (min,+min,max,max)	Invalid (max+1)
Name	4	5 6 19 20	21
Email	11	10 11 29 30	31
Phno	5	6 7 9 10	11
Gender	0	1	2
Age	17	18 19 79 80	81
Password	7	8 9 19 20	21

- **Invalid:**

Name	:	Ram	(value=3)
Email	:	a@gmail.com	(value=11)
Phno	:	91056	(value=5)
Gender	:	“”	(value=0)
Age	:	17	(value=17)
Password	:	aq@2000	(value=7)

- **Valid:**

Name	:	Ashish	(value=6)
Email	:	ashishsv028@gmail.com	(value=21)
Phno	:	7981956235	(value=10)
Gender	:	M	(value=1)
Age	:	19	(value=19)
Password	:	qwer@1234	(value=9)

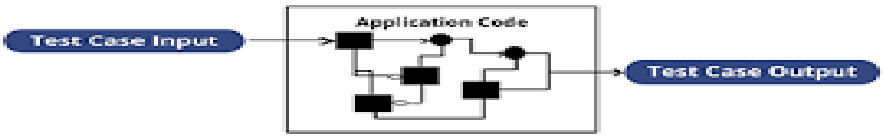
- **Invalid:**

Name	:	Pramodhsairamrangasatyaviswapavan	(value=21)
Email	:	Pramodhramrangasatyaviswapavan@gmailyahoo.com	(value=31)
Phno	:	78012560441	(value=11)
Gender	:	MA	(value=2)
Age	:	100	(value=100)
Password	:	thepasswordispassword@password	(value=25)

2) White box testing:

White-box testing is a method of software testing that tests internal structures or workings of an application, as opposed to its functionality. In white-box testing an internal perspective of the system, as well as programming skills, are used to design test cases.

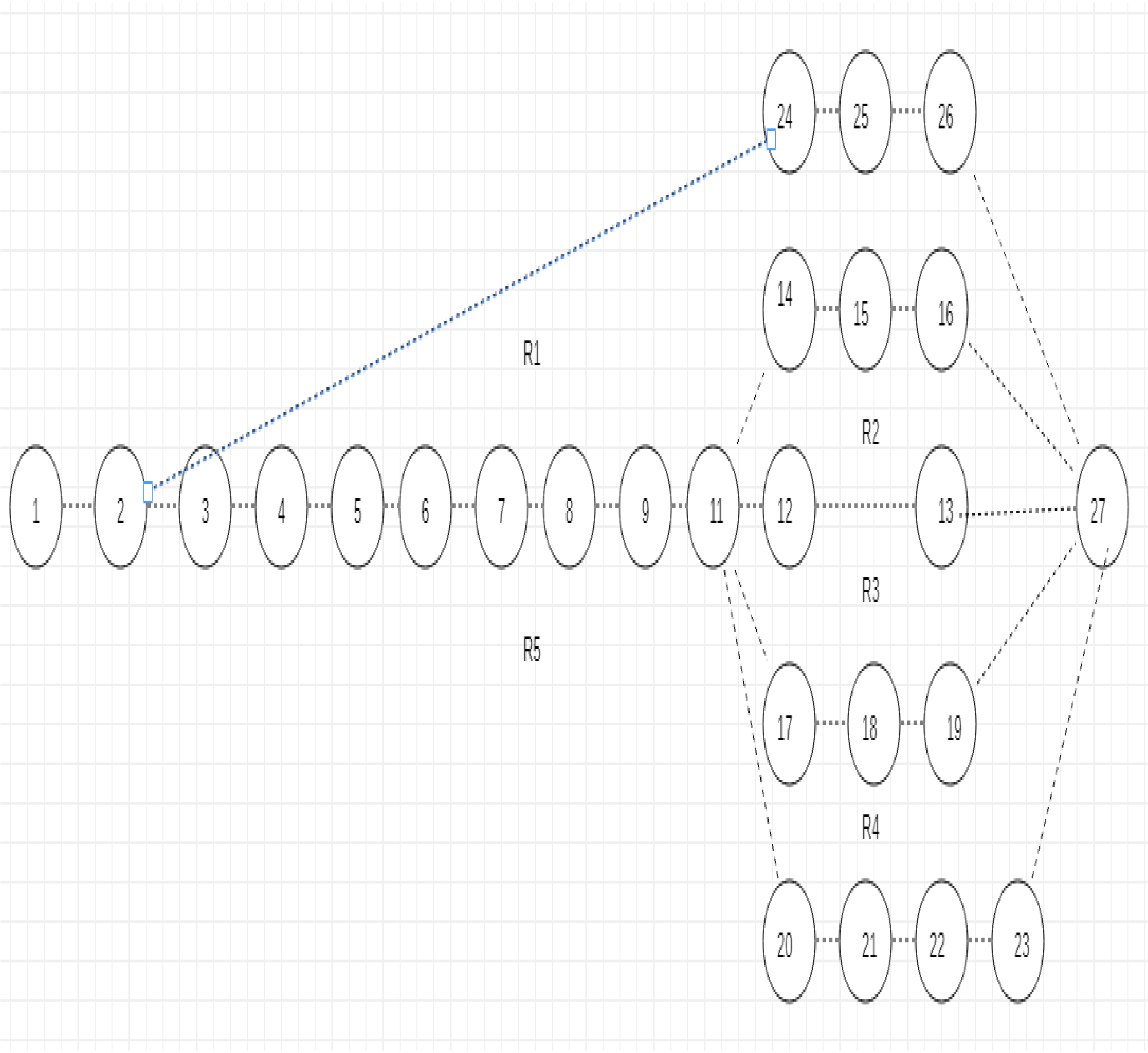
WHITE BOX TESTING APPROACH



Python code for Registration:

```
@app.route('/register',methods=['POST','GET'])
def register():

details=[ i for i in request.form.values()]

if(len(details)>=1):
    name=details[0]
    mail=details[1]
    phno=details[2]
    gender=details[3]
    age=details[4]
    place=details[5]
    pwd=details[6]
    cpwd=details[7]
    if(len(pwd)<=8):
        msg="Password should be more than 8 characters"
        return  render_template("Registration.html",msg=msg)
    elif(len(phno)!=10):
        msg="phone number should be of 10 digits"
        return render_template("Registration.html", msg=msg)
    elif(pwd!=cpwd):
        msg="passwords doesn't matched"
        return render_template("Registration.html", msg=msg)
    else:
        db.db.collection.insert_one({
            "name": details[0],
            "email":details[1],
            "phno":details[2],
            "gender":details[3],
            "age":details[4],
            "place":details[5],
            "pwd":details[6]
        })
        msg="registered successfully"
        return render_template("login.html",msg=msg)

else:

        print(“get request”)
        return render_template("Registration.html")

return  render_template("Registration.html")
```

Control Flow Graph for above code:



Now we need to find the cyclomatic complexity because it is equal to the number of independent paths.

There are total 5 regions in the control flow graph

- ➢ Cyclomatic complexity = predicate nodes + 1

4+1=5

Therefore, we have 5 independent paths implies we need to develop 5 testcases to cover all the paths.

Testcases:

- **TestCase 1:** 1->2->24->25->26->27
  Request Method= "GET"
  Redirects to "Registration.html"

- **TestCase 2:** 1->2->3->4->5->6->7->8->9->11->12->13->27
  Request Method = "POST"
  When user enters password less than 8 characters
  Redirects to "Registration.html"

- **TestCase 3:** 1->2->3->4->5->6->7->8->9->11->20->21->22->23->27
  Request Method = "POST"
  When user enters all details correctly, his doc is created
  Redirects to "Login.html"

- **TestCase 4:** 1->2->3->4->5->6->7->8->9->11->17->18->19->27
  Request Method = "POST"
  When password doesn't matches with confirm password
  Redirects to "Registration.html"

- **TestCase 5:** 1->2->3->4->5->6->7->8->9->11->14->15->16->27
  Request Method = "POST"
  When phone number entered is not of 10 digits
  Redirects to "Registration.html"

White box testing through condition testing/branch testing for signup module:

Condition testing is a test construction method that focuses on exercising the logical conditions in a program module. Errors in conditions can be due to:

- Boolean operator error
- Boolean variable error
- Boolean parenthesis error
- Relational operator error
- Arithmetic expression error

Definition:

For a compound condition C, the true and false branches of C and every simple condition in C need to be executed at least once.

Decision	Possible Outcomes	Test Cases				
		1	2	3	4	5
Method==”POST”	T	X	X	X	X	
	F					X
pwd==cpwd	T	X		X	X	
	F		X			X
Length(phno)==10	T	X	X		X	
	F			X		X
Length(pwd)>=8	T	X	X	X		
	F				X	X

Test Cases		
Case No	Input values	Expected Outcomes

1)	Request Method =”POST”
	Phno=7819019221
	Pwdln=13
	Pwd=qwerty1234567
	Cpwd= qwerty1234567
		Msg: success
		Page: Login

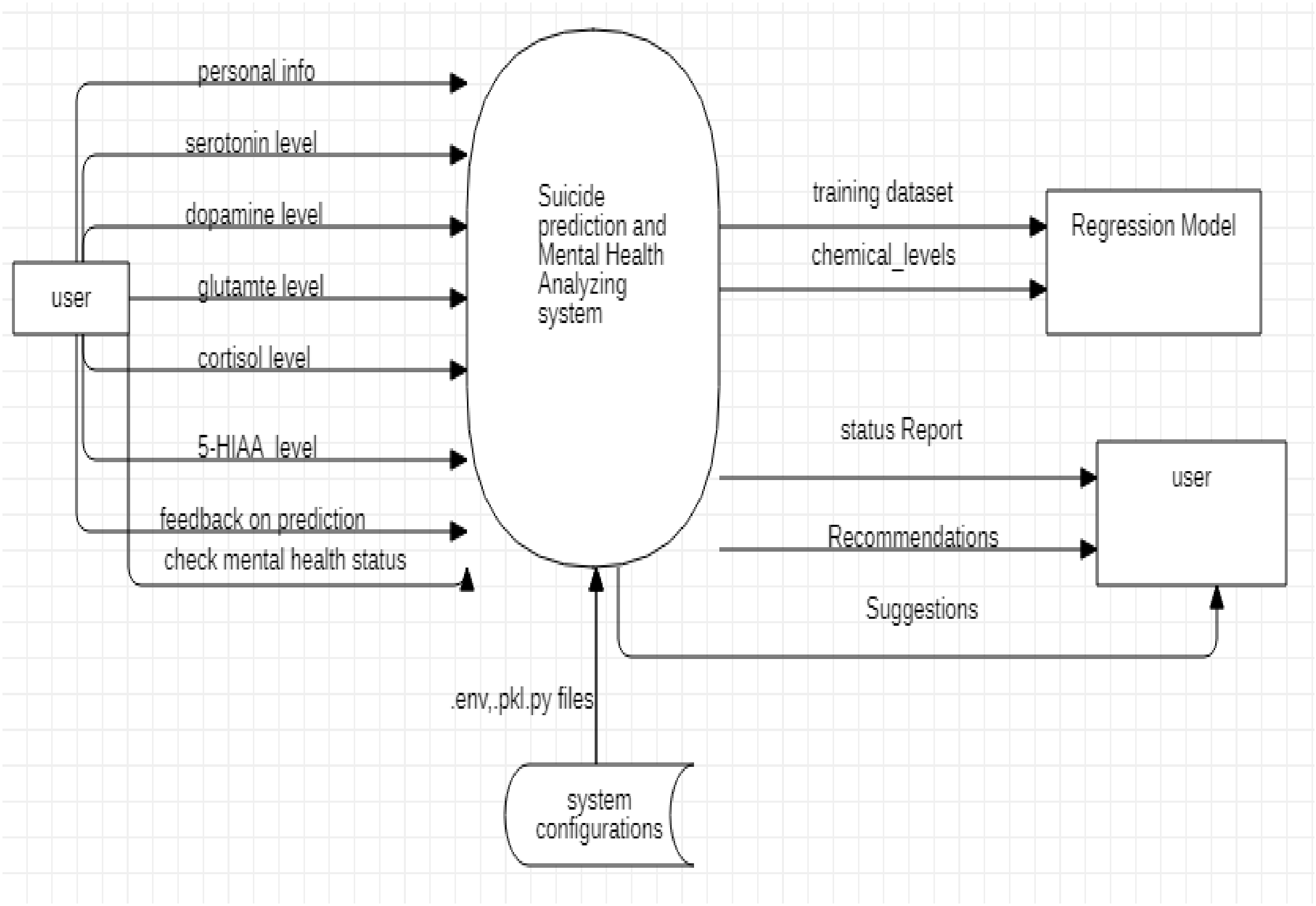
2)	Request Method =”POST”
	Phno=7819019332
	Pwdln=12
	Pwd=qwerty123456
	Cpwd=qwerty1234567
		Msg : pwd and cpwd should be same
		Page :Registration

3)	Request Method =”POST”
	Phno=891984522
	Pwdln=10
	Pwd=qwerty1234
	Cpwd=qwerty1234
		Msg : phone number should have 10 digits
		Page :Registration

4)	Request Method =”POST”
	Phno=9493324135
	Pwdln=6
	Pwd=qwerty
	Cpwd=qwerty
		Msg : pwd length should be minimum 8 characters
		Page :Registration

5)	Request Method =”GET” (prompts to enter details)	Msg: Fill details Page: Registration

EMPIRICAL ESTIMATION:



- **Number of user inputs** = 7 (Personal details, Serotonin levels, dopamine level, Glutamate level, Cortisol level, 5-HIAA level, Feedback)

- **Number of user outputs** = 3 (Status Report, Recommendations, Suggestions)

- **Number of user inquiries** = 1 (Check Mental Health Status)

- **Number of files** = 1 (system configuration file)

- **Number of external interfaces** = 2 (training dataset (.pklfile), chemical_levels)

We calculate Functional Points (F.P's) for average case:

Parameter	Count	Simple	Average	Complex	Total (Count*Avg)
Number of User Inputs	7	3	7	12	7*7=49
Number of User Outputs	3	2	6	11	3*6=18
Number of User Inquiries	1	2	6	11	1*6=6
Number of Files	1	1	7	12	1*7=7
Number of external interfaces	2	4	9	13	2*9=18
Total					=98

F.P Estimation:

- To calculate the total FP, the following formula is used:

FP = CT * (0.65 + 0.01 * ΣFi)

Where:

- FP : Total Function Points
- CT : Count Total
- Σ Fi: Total complexity adjustment or Total Weighting Factor value

Assume Weighting Factor is simple:

CT = TOTAL COUNT = 98

Now we need to find Σ Fi,

General System Characteristics	**Degree of Influence (Value)**
Operational Ease	2
Data Communication	5
Distributed Functions	3
Performance	5
Heavily Used Configuration	3
Transaction Rate	5
On-line Data Entry	3
On-line Update	4
End-user Efficiency	5
Complex Processing	1
Reusability	4
Installation Ease	1
Multiple Sites	2
Facilitates Change	4
Total complexity adjustment or Total Weighting Factor value	47

Now, F.P =98 x [0.65 + (0.01 x 47)] = 109.76 ≈ 110 F.Ps

FP=110

LOC Estimation:

To calculate the numbers of line of code, the following formula is used:

LOC = LOC per FP * FP (OR) LOC = AVC * FP

where:

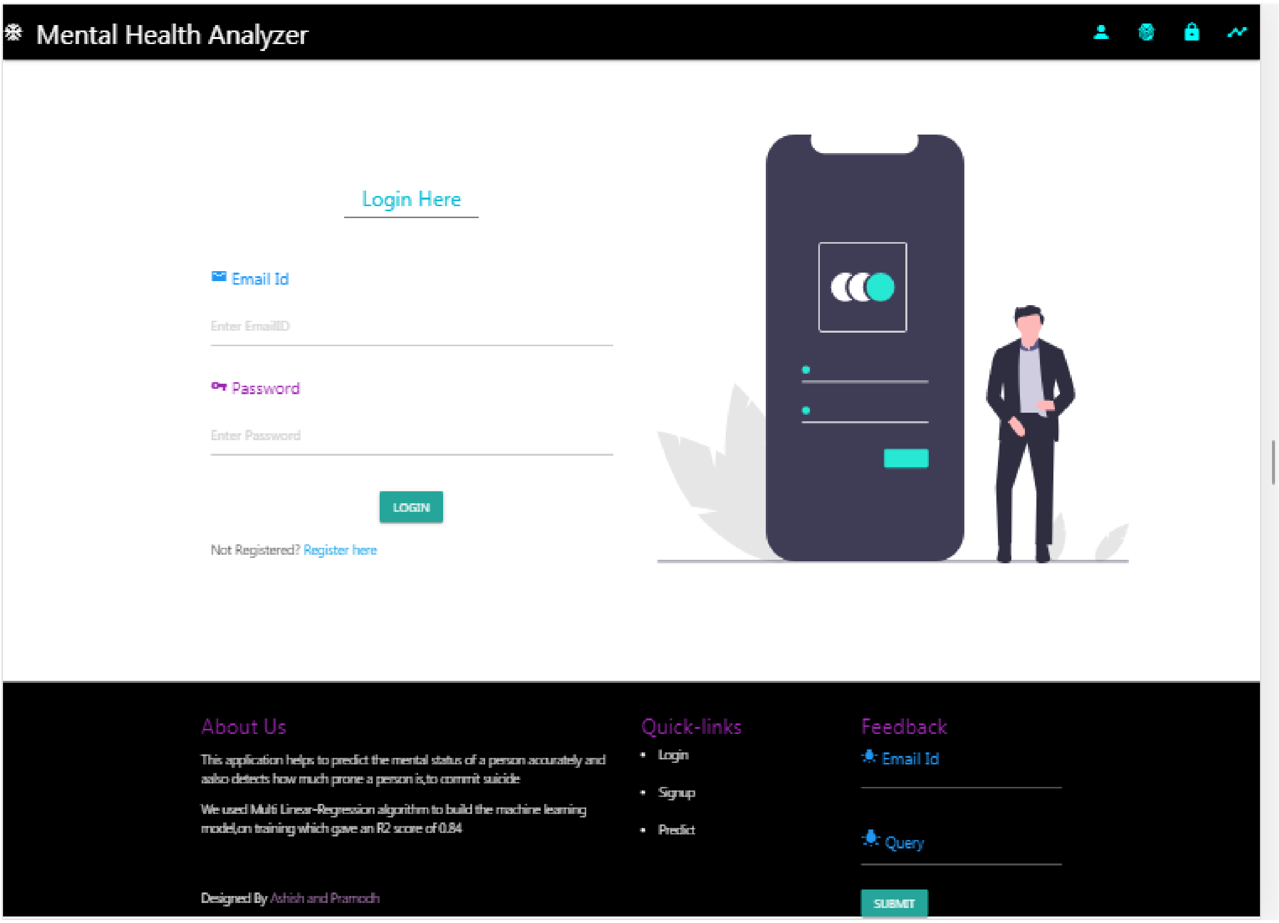
- **AVC** : is the average number of LOC/FP for a given language
- **LOC** : is the numbers of line of code
- **FP** : is the Total numbers of Function Point

Therefore,

LOC = FP * 40 = 110 * 40 = **4400 LOCs**

GUI Interface Screenshots:

LOGIN INTERFACE:

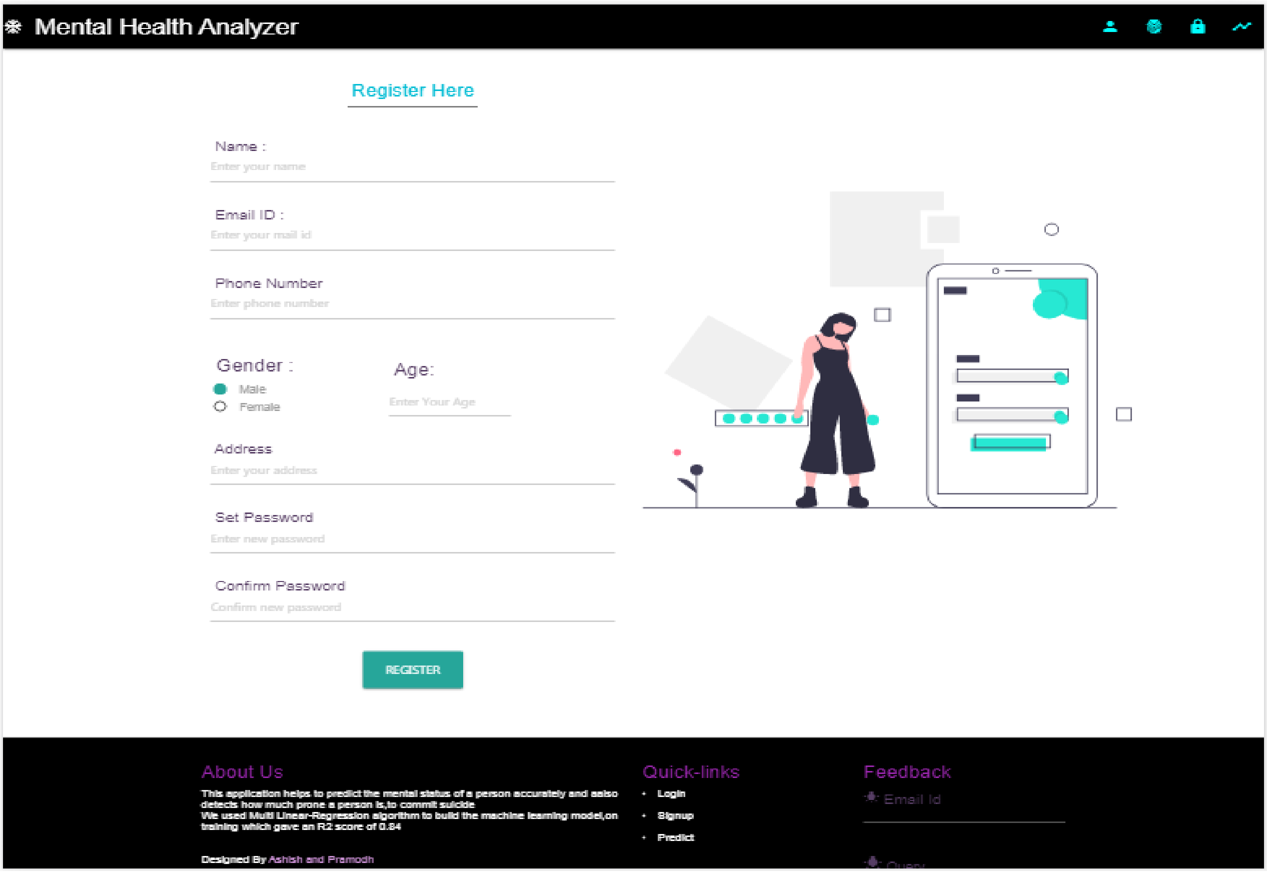


This application helps to predict the mental status of a person accurately and aalso detects how much prone a person is,to commit suicide

We used Multi Linear-Regression algorithm to build the machine learning model,on training which gave an R2 score of 0.84

Designed By Ashish and Pramodh

REGISTER PAGE INTERFACE



Mental Health Analyzer

Register Here

Name :
Enter your name

Email ID :
Enter your mail id

Phone Number
Enter phone number

Gender :
● Male
○ Female

Age:
Enter Your Age

Address
Enter your address

Set Password
Enter new password

Confirm Password
Confirm new password

REGISTER

About Us

This application helps to predict the mental status of a person accurately and aalso detects how much prone a person is,to commit suicide
We used Multi Linear-Regression algorithm to build the machine learning model,on training which gave an R2 score of 0.84

Designed By Ashish and Pramodh

Quick-links

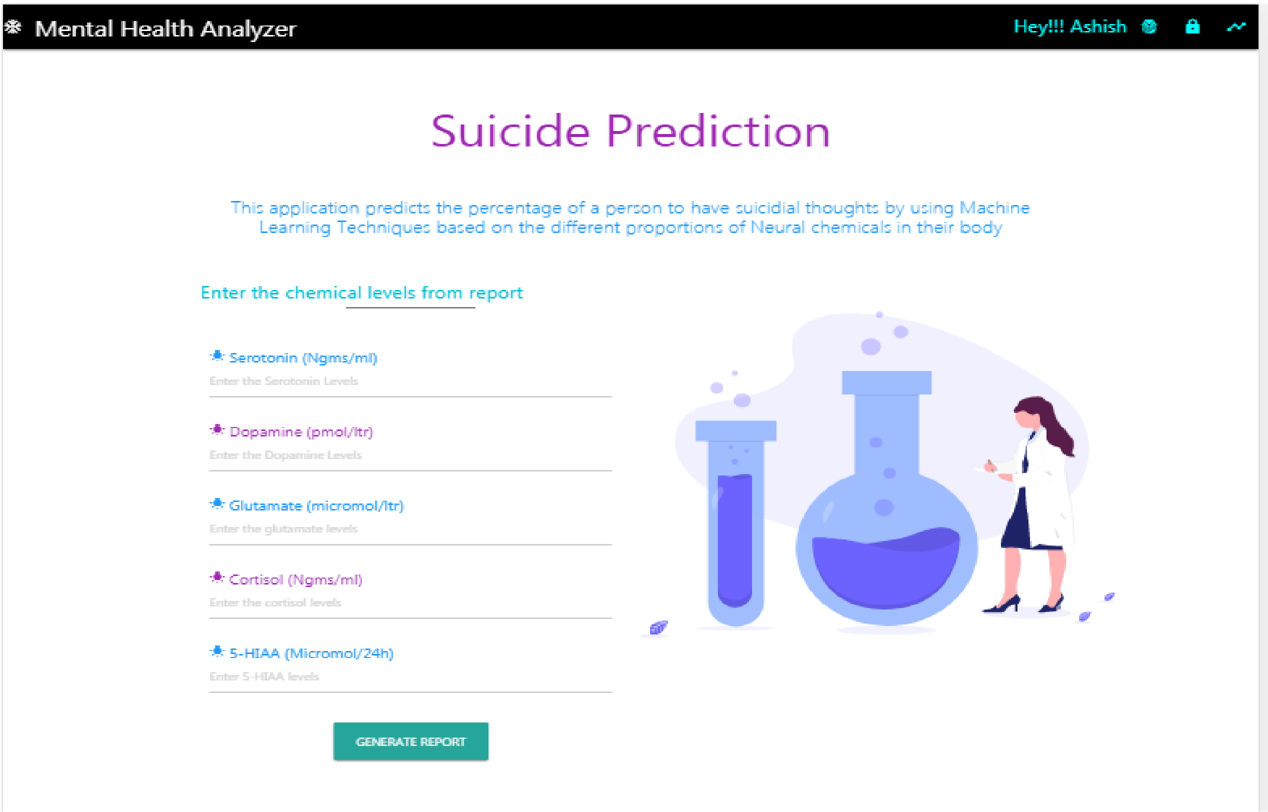
- Login
- Signup
- Predict

Feedback

Email Id

Query

PREDICTIONS PAGE INTERFACE:



Mental Health Analyzer

Hey!!! Ashish

Suicide Prediction

This application predicts the percentage of a person to have suicidial thoughts by using Machine Learning Techniques based on the different proportions of Neural chemicals in their body

Enter the chemical levels from report

Serotonin (Ngms/ml)
Enter the Serotonin Levels

Dopamine (pmol/ltr)
Enter the Dopamine Levels

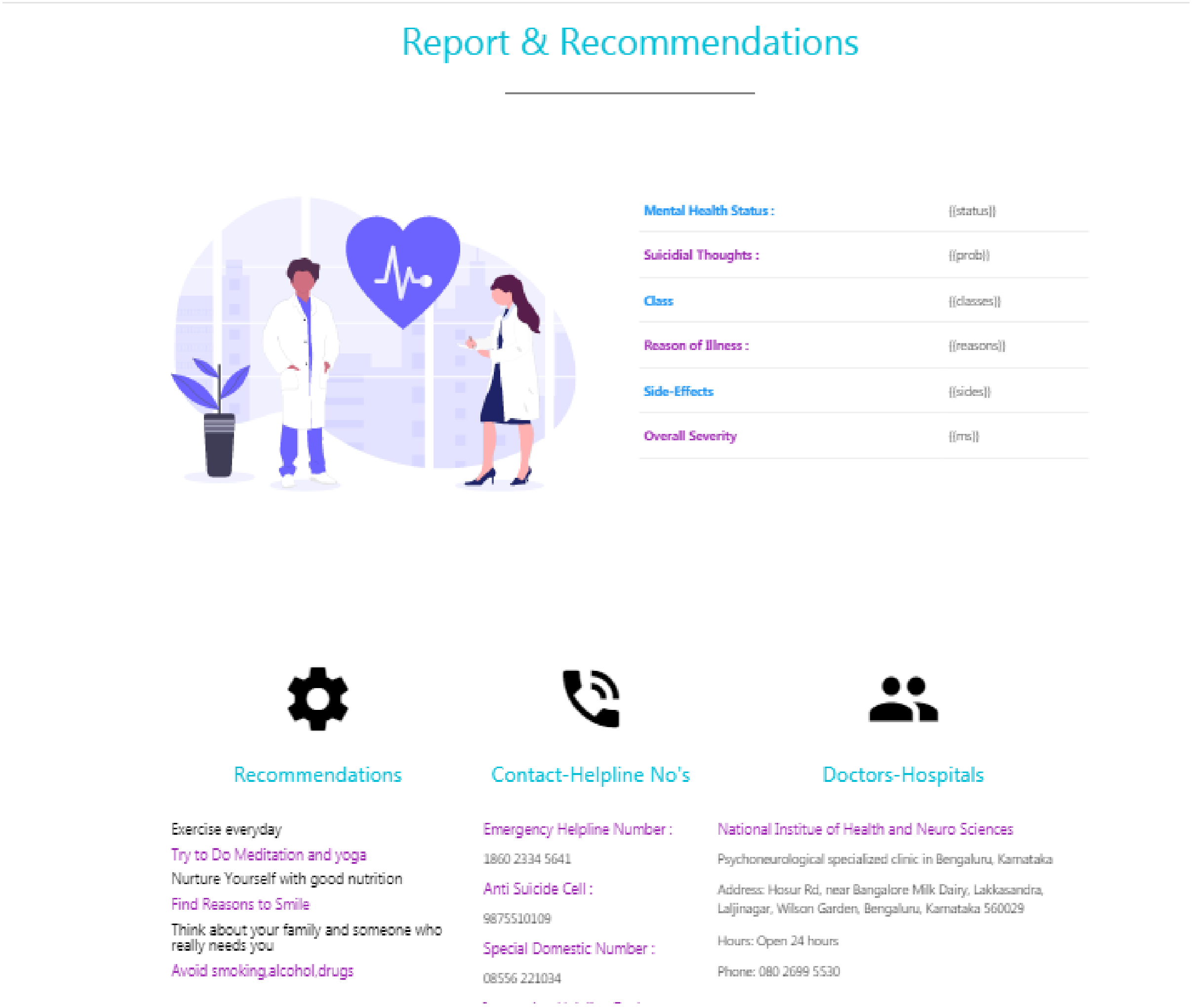
Glutamate (micromol/ltr)
Enter the glutamate levels

Cortisol (Ngms/ml)
Enter the cortisol levels

5-HIAA (Micromol/24h)
Enter 5-HIAA levels

GENERATE REPORT

REPORTS AND RECOMMENDATIONS INTEERFACE



Report & Recommendations

Mental Health Status :	{{status}}
Suicidial Thoughts :	{{prob}}
Class	{{classes}}
Reason of Illness :	{{reasons}}
Side-Effects	{{sides}}
Overall Severity	{{rms}}

Recommendations

Exercise everyday
Try to Do Meditation and yoga
Nurture Yourself with good nutrition
Find Reasons to Smile
Think about your family and someone who really needs you
Avoid smoking,alcohol,drugs

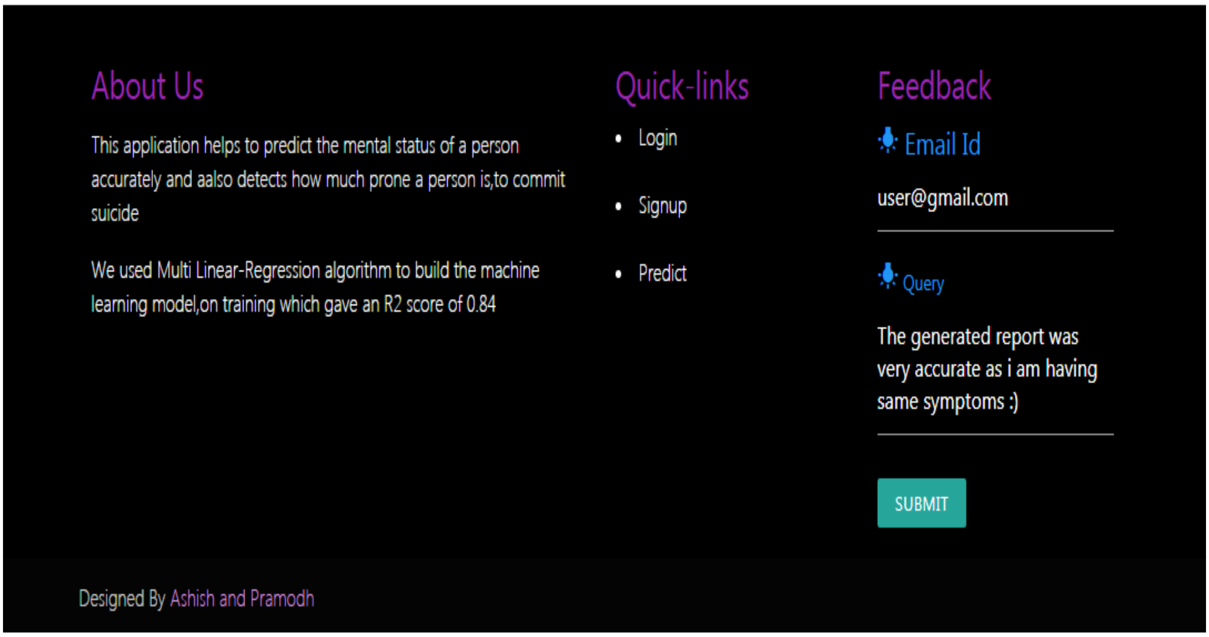
Contact-Helpline No's

Emergency Helpline Number :
1860 2334 5641
Anti Suicide Cell :
9875510109
Special Domestic Number :
08556 221034

Doctors-Hospitals

National Institue of Health and Neuro Sciences
Psychoneurological specialized clinic in Bengaluru, Karnataka
Address: Hosur Rd, near Bangalore Milk Dairy, Lakkasandra, Laljinagar, Wilson Garden, Bengaluru, Karnataka 560029
Hours: Open 24 hours
Phone: 080 2699 5530

FEEDBACK AND ABOUT US INTERFACE



About Us

This application helps to predict the mental status of a person accurately and aalso detects how much prone a person is,to commit suicide

We used Multi Linear-Regression algorithm to build the machine learning model,on training which gave an R2 score of 0.84

Quick-links

- Login
- Signup
- Predict

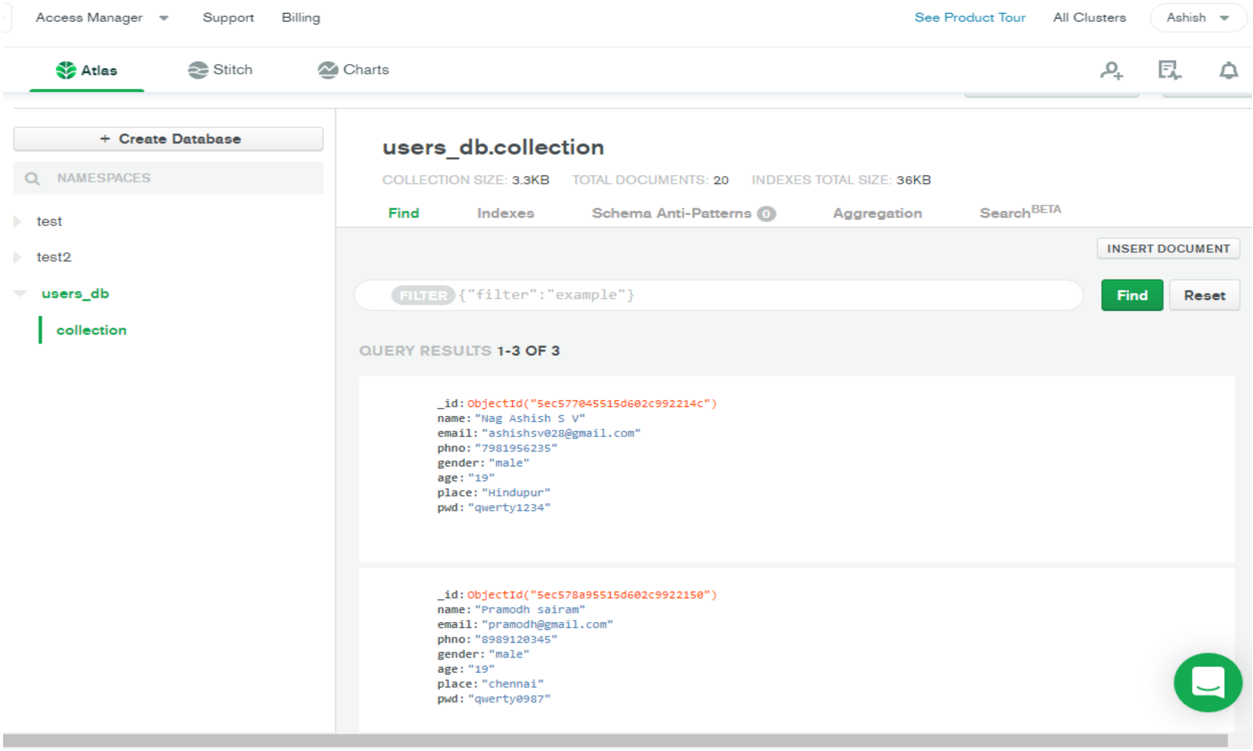
Feedback

Email Id
user@gmail.com

Query
The generated report was very accurate as i am having same symptoms :)

Designed By Ashish and Pramodh

USERS_DATABASE:



users_db.collection

COLLECTION SIZE: 3.3KB TOTAL DOCUMENTS: 20 INDEXES TOTAL SIZE: 36KB

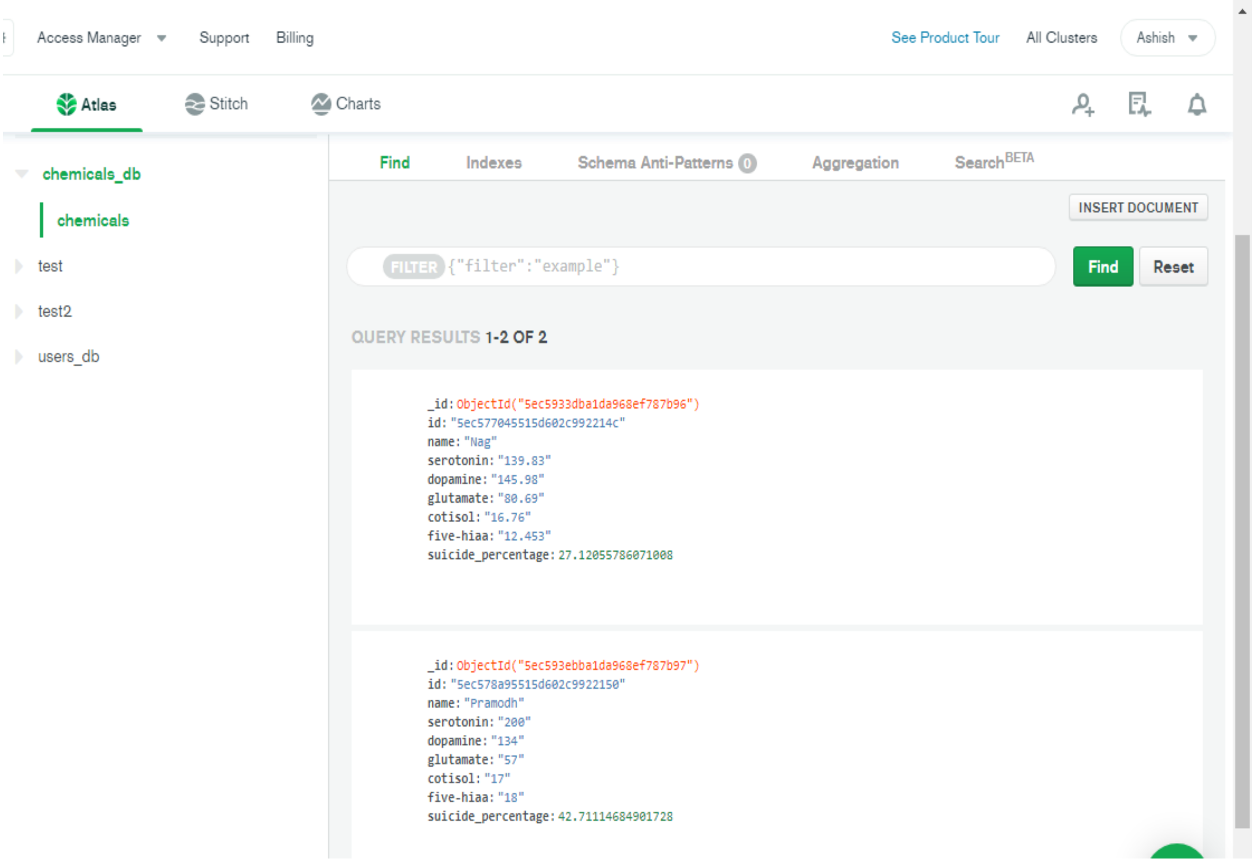
{"filter":"example"}

QUERY RESULTS 1-3 OF 3

```
_id: ObjectId("5ec577045515d602c992214c")
name: "Nag Ashish S V"
email: "ashishsv028@gmail.com"
phno: "7981956235"
gender: "male"
age: "19"
place: "Hindupur"
pwd: "qwerty1234"
```

```
_id: ObjectId("5ec578a95515d602c9922150")
name: "Pramodh sairam"
email: "pramodh@gmail.com"
phno: "8989120345"
gender: "male"
age: "19"
place: "chennai"
pwd: "qwerty0987"
```

CHEMICALS_DATABASE:



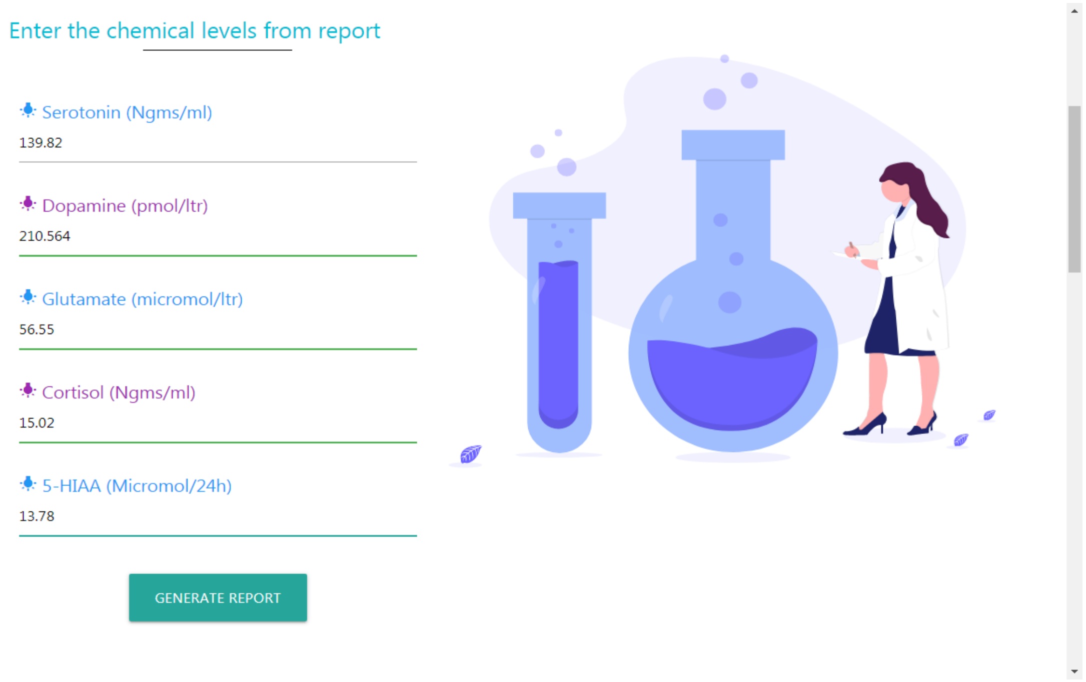
{"filter":"example"}

QUERY RESULTS 1-2 OF 2

```
_id: ObjectId("5ec5933dba1da968ef787b96")
id: "5ec577045515d602c992214c"
name: "Nag"
serotonin: "139.83"
dopamine: "145.98"
glutamate: "80.69"
cotisol: "16.76"
five-hiaa: "12.453"
suicide_percentage: 27.12055786071008
```

```
_id: ObjectId("5ec593ebba1da968ef787b97")
id: "5ec578a95515d602c9922150"
name: "Pramodh"
serotonin: "200"
dopamine: "134"
glutamate: "57"
cotisol: "17"
five-hiaa: "18"
suicide_percentage: 42.71114684901728
```

INPUT:



Enter the chemical levels from report

Serotonin (Ngms/ml)
139.82

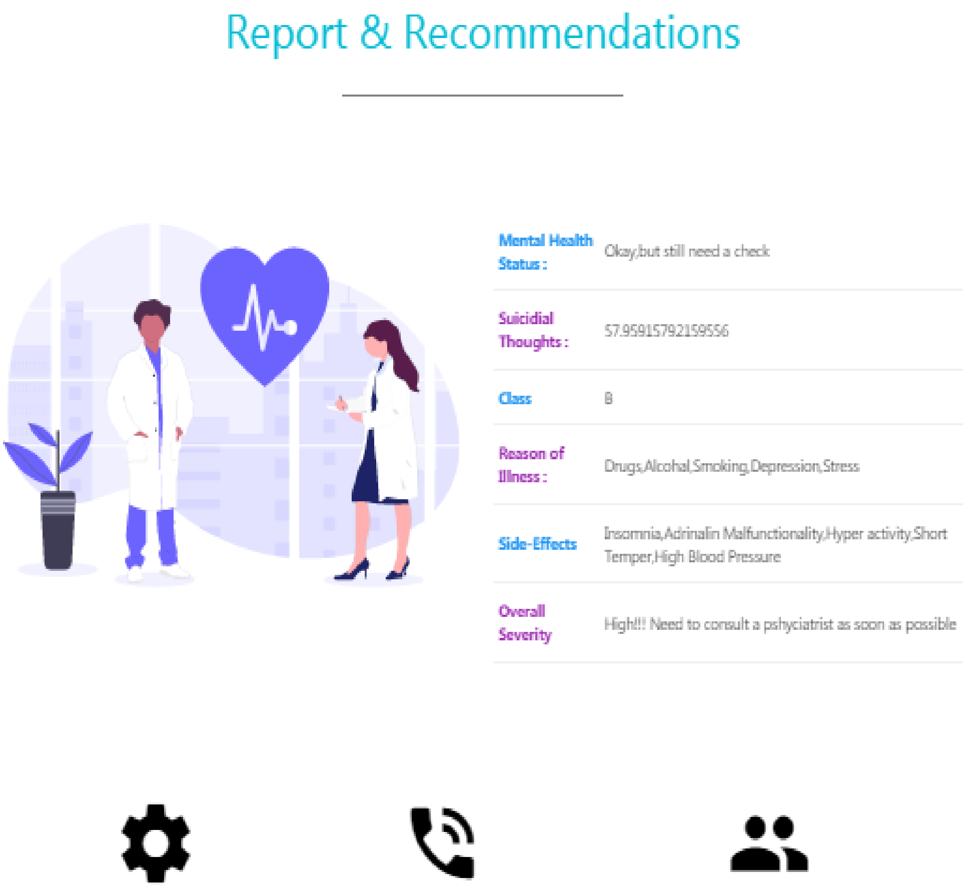
Dopamine (pmol/ltr)
210.564

Glutamate (micromol/ltr)
56.55

Cortisol (Ngms/ml)
15.02

5-HIAA (Micromol/24h)
13.78

OUTPUT:



Report & Recommendations

Mental Health Status :	Okay,but still need a check
Suicidial Thoughts :	57.95915792159556
Class	B
Reason of Illness :	Drugs,Alcohal,Smoking,Depression,Stress
Side-Effects	Insomnia,Adrinalin Malfunctionality,Hyper activity,Short Temper,High Blood Pressure
Overall Severity	High!!! Need to consult a pshyciatrist as soon as possible