

Chapter-1

Introduction

Suicide death occurs when a person intentionally ends their own life. Suicide is the 10th leading cause of death among Americans and is the 9th leading cause of death among Tennesseans. For young Tennessee residents aged 10 to 24 years of age, suicide is the 3rd leading cause of death and it is the 2nd leading cause of death for adults aged 25-64. However, suicide deaths are only part of the problem. More people survive suicide attempts than die. They are often seriously injured and need medical care.

Suicide is a serious public health issue and an immeasurable tragedy for the surviving families, friends, and communities. Suicide is often related to mental care.

Types of suicide:

According to Emile Durkheim's (A Study in Sociology, New York, The Free Press, 1951) theory on suicide, he concluded that there are four different types of suicide. Durkheim's studies revolved around him trying to understand what makes a person actually commit suicide and what influences or factors may have led that person to that final decision or act.

Durkheim believed that various sociologically factors and influences were at work such as work pressure, financial, religious, marital to name just a few. According to Durkheim, there are four types of suicide: Egoistic suicide, Altruistic suicide, Anomic suicide, Fatalistic suicide.

Causes of suicide:

1. Family history of suicide
2. Family history of child maltreatment
3. Previous suicide attempt(s)
4. History of alcohol and substance abuse
5. Feelings of hopelessness
6. Impulsive or aggressive tendencies
7. Cultural and religious beliefs
8. Local epidemics of suicide
9. Isolation, a feeling of being cut off from other people
10. Barriers to accessing mental health treatment
11. Loss (relational, social, work, or financial)
12. Physical illness
13. Easy access to lethal methods suicide
14. Mental Stress
15. Financial Problem
16. Mental Illnesses
17. Serious illness such as chronic pain
18. Criminal/legal problem
19. Emotional or aggressive tendencies
20. Loss of relationships
21. Victim of bullying
22. Victim of discrimination
23. Victim of blackmail
24. Violence victimization and perpetration
25. Social isolation
26. Suicide clusters in communities

1.1 Motivation

Suppose, you don't feel good about anything. Then you can check your suicidal tendency. If tendency high / 60% up, then you need mental support and doctor consultant. Where can you find a doctor? We site provide predict result and doctor info for your need.

Suppose, you doctor, you can't guess suicidal tendency of your patient. No Problem you visit predictuse.com and quickly predict. You get proper prediction.

Suppose, you need employee for your company. You can check suicidal tendency of your new employee.

Our system also has many functional uses. Login now to get service from our system. **predictuse.com**

1.2 Objectives

1. To predict someone's suicidal tendency observing their activities.
2. To maintain stable mental condition of suicidal prone people.
3. To provides some medical trips and mental support for users.
4. The company can check mental condition of employee judgment.

Chapter 2

Background

In children and adolescents with a depressive disorder, predicting who will also go on to exhibit suicide-related behaviors (SRBs), including suicide attempt or self-harm, is a key challenge facing clinicians. Aims: To investigate the relative contributions of depressive disorder severity, hopelessness, family dysfunction, and perceived social support to the risk of suicide-related behaviors. Methods: This was a cross-sectional study of a group of 10–16-year-olds with major depressive disorders and dysthymic disorder. Results: Child-rated depressive disorder symptom severity emerged as the greatest predictor of risk. Hopelessness and family dysfunction were also significant predictors of SRBs. In combination these variables were strong predictors, accounting for 66% of the variance. This is a cross-sectional study design, rather than longitudinal, therefore risk prediction over time was not possible. Conclusions: Understanding the child and adolescents depressive disorder symptom severity from their perspective, their level of hopelessness, as well as their family context is critical in understanding the risk of SRBs. These findings may help to provide direction for targeted interventions to address these clinical risk factors. We conducted a review combining articles and abstracts with full thesis. We searched PubMed, PubMed Central, Google Scholar, Science Direct and, google using multiple terms related to suicide without any date boundary and without any basis of types of studies, that is, all types of studies were scrutinized. Finally, 16 articles were selected for review. Report suggested that every day almost 32 people commit suicide in 2019 which was 29 and 30 in 2015 and 2017 respectively. The mortality rate of suicide found 39.6 per 100,000 in Bangladesh. The most common method is hanging followed by poisoning and jumping under the train. The most

prevalent age group is age under 40 years. The rate of suicide in children is also increasing. In contrast to most Asian countries, more Bangladeshi women commit suicide than men. The mean age of male and female were 28.86 ± 11.27 years and 25.31 ± 7.70 years respectively. The most common associated factors of suicide are younger age, lower education, students, nuclear family, family history of suicide, use substance, problem in workplace, financial constraints, affair, domestic violence, divorce, and physical illness. Most of the suicidal event occurred at night, followed by morning (6 am–12 am), and evening. It's a criminal offence in Bangladesh. The source of information is mainly police, forensic reports, media and courts.

2.1 Symptoms

- Talking about suicide — for example, making statements such as "I'm going to kill myself," "I wish I were dead" or "I wish I hadn't been born".
- Getting the means to take your own life, such as buying a gun.
- Withdrawing from social contact and wanting to be left alone.
- Being preoccupied with death, dying or violence.
- Feeling trapped or hopeless about a situation.
- Increasing use of alcohol or drugs.
- Changing normal routine, including eating or sleeping patterns.
- Doing risky or self-destructive things, such as using drugs or driving recklessly
- Saying goodbye to people as if they won't be seen again.

Chapter 3

Related Works

Previous research indicates an individual's most intense instance of suicidal ideation, referred to as worst-point suicidal ideation, is an especially strong predictor of suicide attempts and death by suicide (Beck et al., 1999; Law, Jin, Anestis, 2018; Nam, Hilimire, Jahn, Lehmann, & DeVylder, 2018; Nock et al., 2018). Larger deviations in suicidal ideation over the course of outpatient mental health treatment have also been observed among patients with a history of multiple suicide attempts (Bryan & Rudd, 2016), a clinical subgroup with deficits in self-regulatory capacity. Worst-point suicidal ideation may therefore be an indicator of self-regulation. The ability to self-regulate in response to stressful situations theoretically explains why different situations and stressors lead to suicidal crises in some individuals but not others (e.g., Beck & Haigh, 2014; Rudd, 2006).

Advances in artificial intelligence and ubiquitous computing opened possibilities for researches in health areas (Bavaresco et al., 2020; Dias et al., 2020; Paula et al., 2021; Tavares & Barbosa, 2020; Vianna & Barbosa, 2020). The application of these technologies in suicide science received the attention of researchers in the last years. The studies focused on different domains, such as suicidal ideation, behavior, thoughts, and self-harm. The approaches can combine data from different sources.

Chapter 4

Implementations

Implementation is the execution or practice of a plan, a method or any design, idea, model, specification, standard or policy for doing something.

4.1 ML Model

Machine Learning models can be understood as a program that has been trained to find patterns within new data and make predictions. These models are represented as a mathematical function that takes requests in the form of input data, makes predictions on input data, and then provides an output in response. First, these models are trained over a set of data, and then they are provided an algorithm to reason over data, extract the pattern from feed data and learn from those data. Once these models get trained, they can be used to predict the unseen dataset.

Brain.js provides multiple neural network implementations as different neural nets can be trained to do different things well.

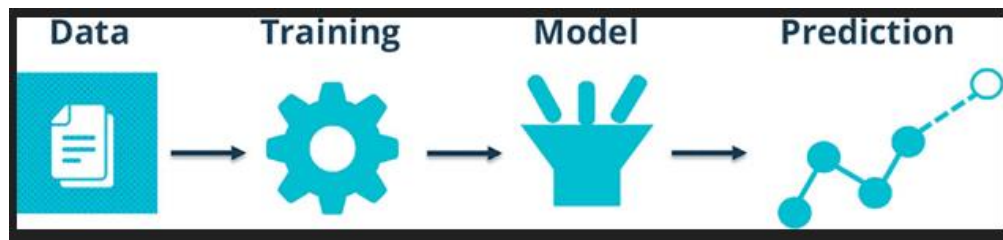


Fig 1 : ML Model

4.2 Project Model

REST stands for Representational State Transfer. REST is web standards based architecture and uses HTTP Protocol. It revolves around resource where every component is a resource and a resource is accessed by a common interface using HTTP standard methods. REST was first introduced by Roy Fielding in 2000.

A REST Server simply provides access to resources and REST client accesses and modifies the resources using HTTP protocol. Here each resource is identified by URIs/ global IDs. REST uses various representation to represent a resource like text, JSON, XML but JSON is the most popular one.

REST based architecture.

- **GET** – This is used to provide a read only access to a resource.
- **PUT** – This is used to create a new resource.
- **DELETE** – This is used to remove a resource.
- **POST** – This is used to update existing resource or create a new resource.

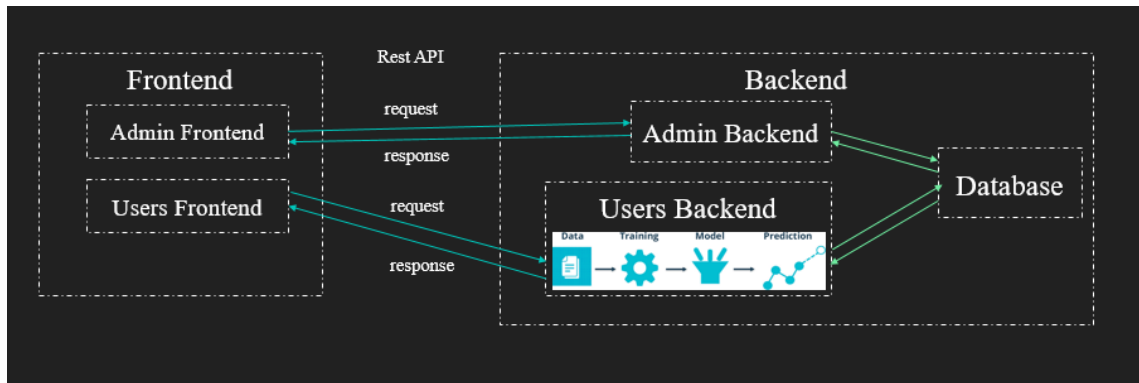


Fig 2 : Project Model

4.3 Visual Studio Code

Visual Studio Code, also commonly referred to as VS Code, is a source-code editor made by Microsoft with the Electron Framework, for Windows, Linux and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git.

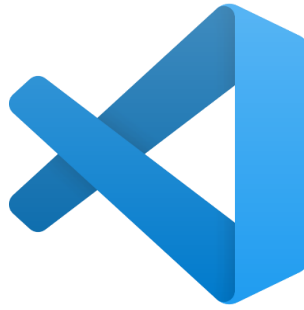


Fig 3 : Visual Studio Code

4.4 Node.js

Node.js is an open-source server environment. Node.js is cross-platform and runs on Windows, Linux, Unix, and macOS. Node.js is a back-end JavaScript runtime environment. Node.js runs on the V8 JavaScript Engine and executes JavaScript code outside a web browser.



Fig 4 : Node.js

4.5 Express.js

Express.js, or simply Express, is a back end web application framework for building RESTful APIs with Node.js, released as free and open-source software under the MIT License. It is designed for building web applications and APIs. It has been called the de facto standard server framework for Node.js.



Fig 5 : Express.js

4.6 Brain.js

Brain.js is a JavaScript library used for neural networking, which is released as free and open-source software under the MIT License. It can be used in both the browser and Node.js backend.

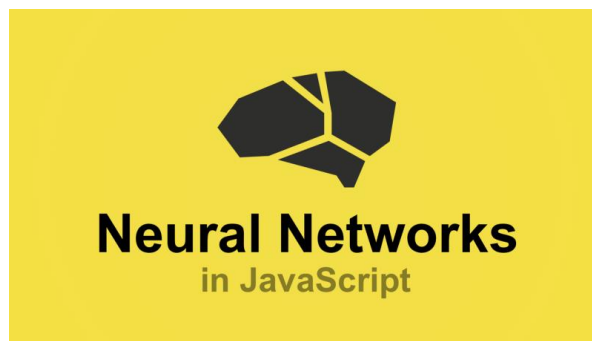


Fig 6 : Brain.js

4.7 Mysql

MySQL is an open-source relational database management system. Its name is a combination of "My", the name of co-founder Michael Widenius's daughter My, and "SQL", the acronym for Structured Query Language



Fig 7 : MySql

4.8 React.js

React is a free and open-source front-end JavaScript library for building user interfaces based on UI components. It is maintained by Meta and a community of individual developers and companies.

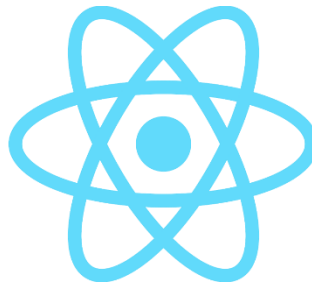


Fig 8 : React.js

4.9 Axios

In a nutshell, Axios is a Javascript library used to make HTTP requests from node.js or XMLHttpRequests from the browser that also supports the ES6 Promise API. Great, so from that we gather it does something that we can already do and that has recently been made significantly better... So why bother?

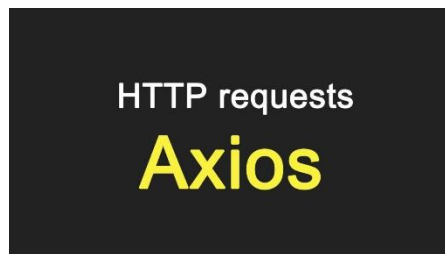


Fig 9 : Axios

4.10 Recharts

Recharts is a Redefined chart library built with React and D3. The main purpose of this library is to help you to write charts in React applications without any pain. Main principles of Recharts are: Simply deploy with React components. Native SVG support, lightweight depending only on some D3 submodules

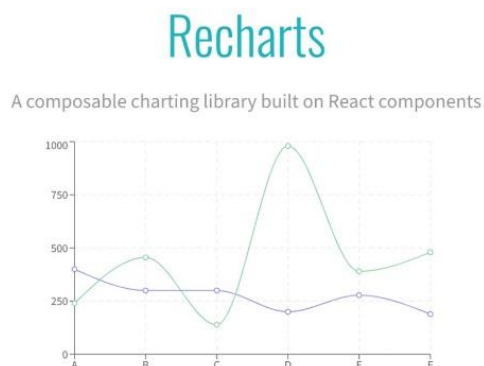


Fig 10 : Recharts

4.11 C-Pannel

cPanel is a web hosting control panel software developed by cPanel, LLC. It provides a graphical interface and automation tools designed to simplify the process of hosting a web site to the website owner or the "end user". It enables administration through a standard web browser using a three-tier structure.

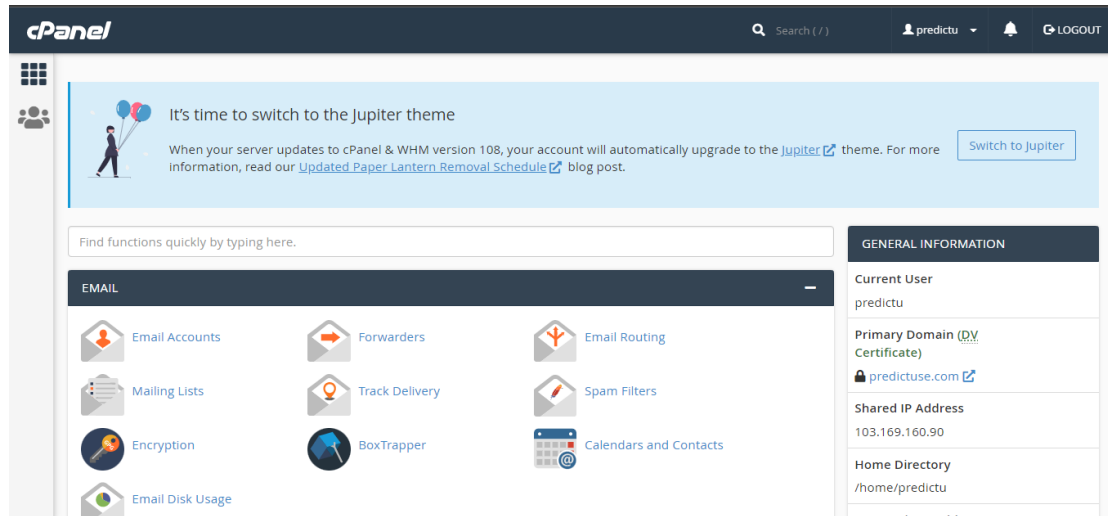


Fig 11 : c-pannel 1

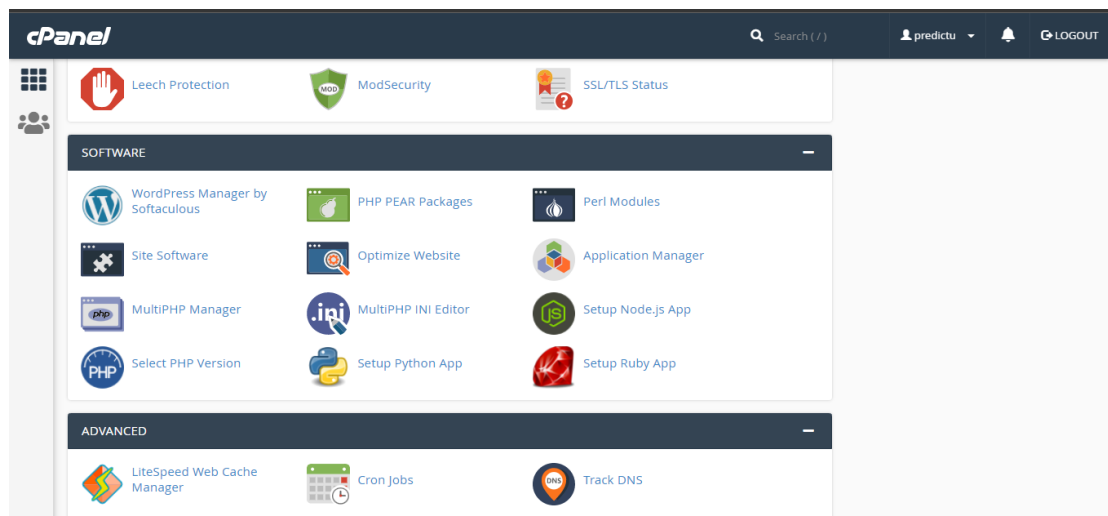


Fig 12 : c-pannel 2

4.12 Backend

- Scripting languages like PHP, Python, Ruby, Perl, Node.js, or Compiled languages like C#, Java or Go
- Automated testing frameworks for the language being used
- Application Data Access
- Application Business Logic
- Database administration
- Scalability
- High availability
- Security concerns, authentication and authorization
- Software Architecture
- Data transformation
- Backup methods and software

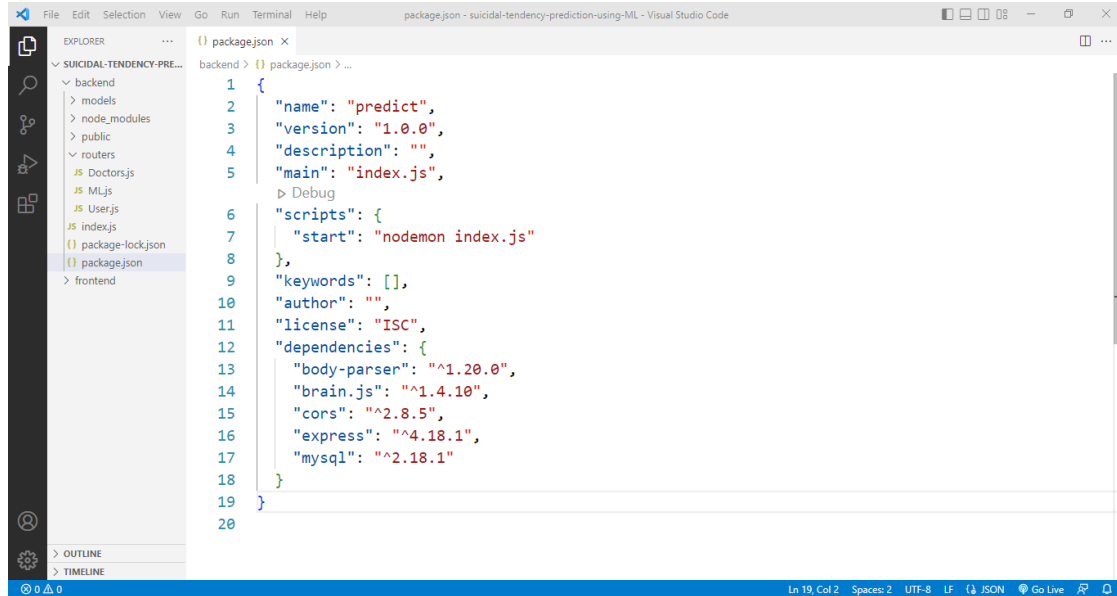
Note that both positions, despite possibly working on one product, have a very distinct set of skills.

API

The frontend communicates with backend through an API. In the case of web and mobile frontends, the API is often based on HTTP request/response. The API is sometimes designed using the "Backend for Frontend" (BFF) pattern, that serves responses to ease the processing on frontend side.

4.13 Backend Code

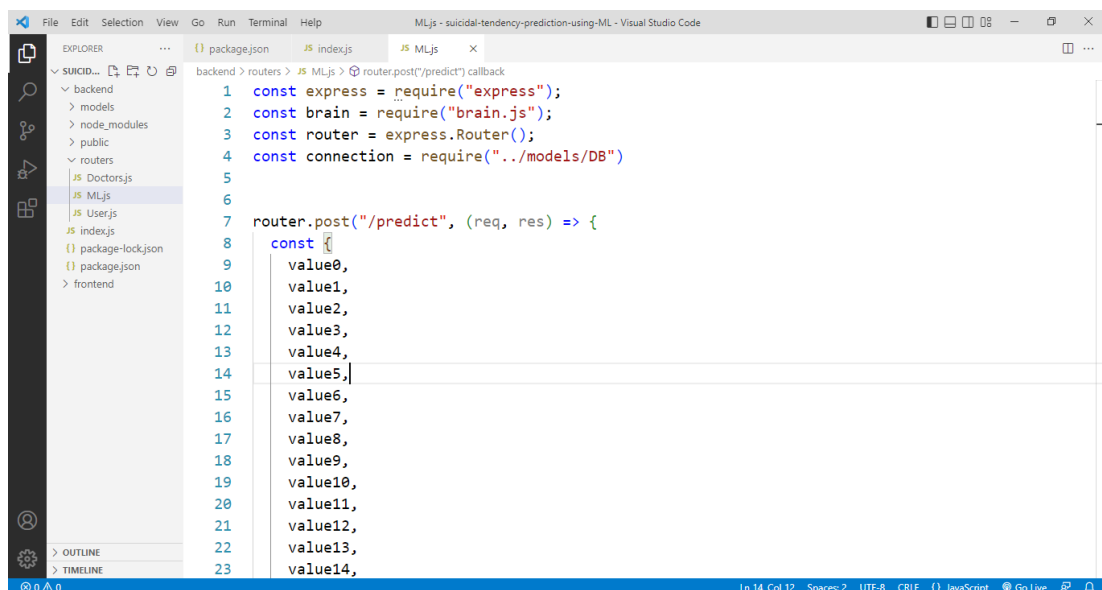
It is Node.js json file. This file includes metadata such as name, author and version as well as starter scripts and dependencies used in the project. Setup node.js server.



```
1 {
2   "name": "predict",
3   "version": "1.0.0",
4   "description": "",
5   "main": "index.js",
6   "scripts": {
7     "start": "nodemon index.js"
8   },
9   "keywords": [],
10  "author": "",
11  "license": "ISC",
12  "dependencies": {
13    "body-parser": "^1.20.0",
14    "brain.js": "^1.4.10",
15    "cors": "^2.8.5",
16    "express": "^4.18.1",
17    "mysql": "^2.18.1"
18  }
19 }
```

Fig 13 : Backend Code 1

It is backend code. It is backend predict router.post route.



```
1 const express = require("express");
2 const brain = require("brain.js");
3 const router = express.Router();
4 const connection = require("../models/DB")
5
6
7 router.post("/predict", (req, res) => {
8   const [
9     value0,
10    value1,
11    value2,
12    value3,
13    value4,
14    value5,
15    value6,
16    value7,
17    value8,
18    value9,
19    value10,
20    value11,
21    value12,
22    value13,
23    value14,
```

Fig 14 : Backend Code 2

Creating a REST API backend using node.js, express.js. This code frontend input data receive and manae.

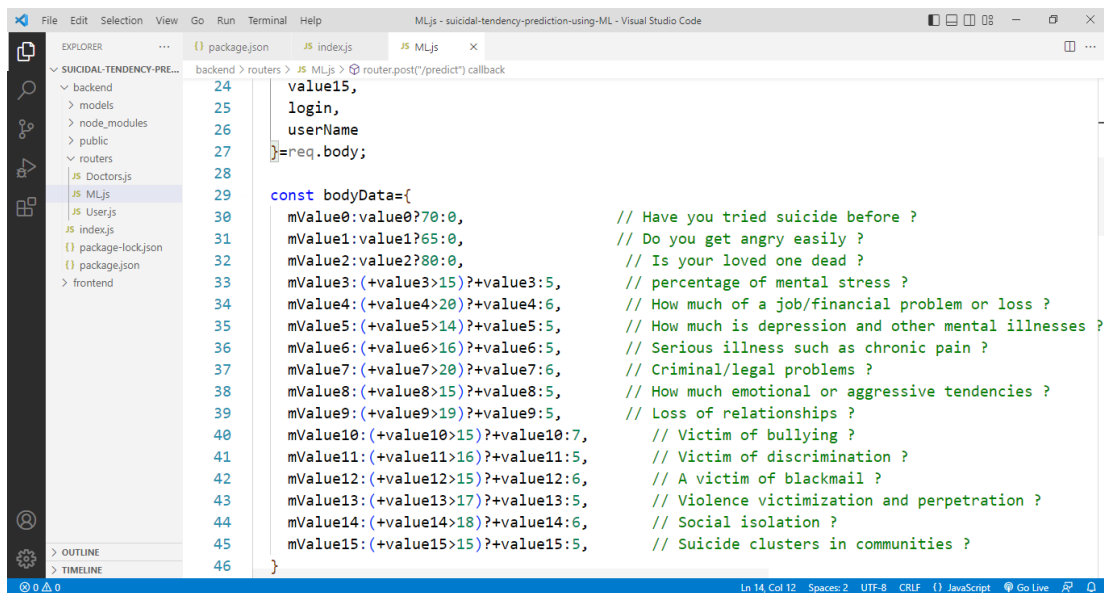


Fig 15 : Backend Code 3

This code is valid user check. If true then data get from mysql database.

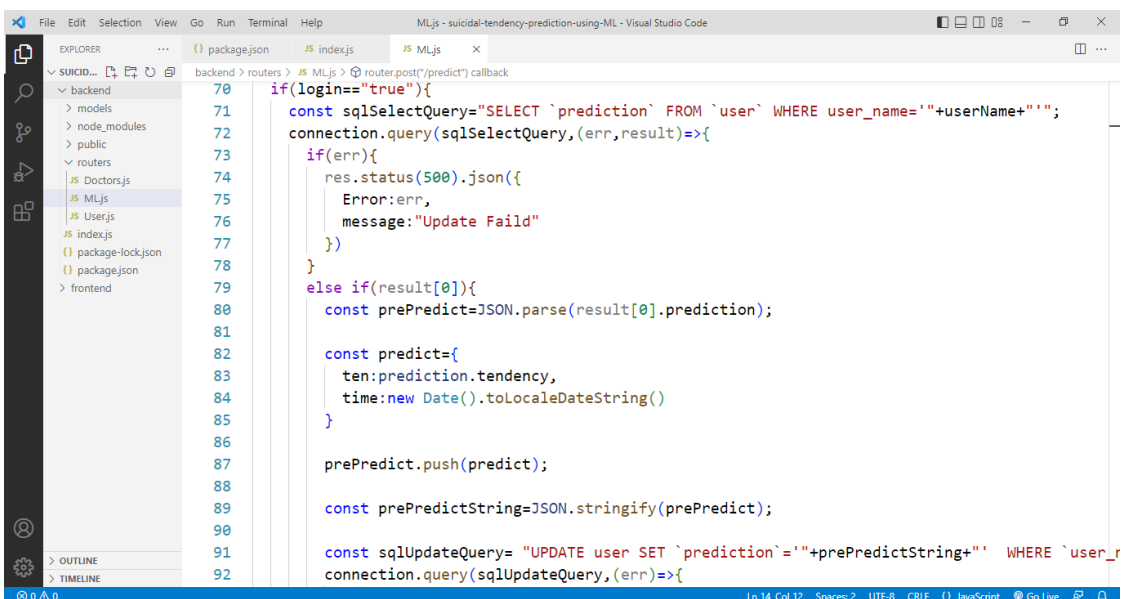
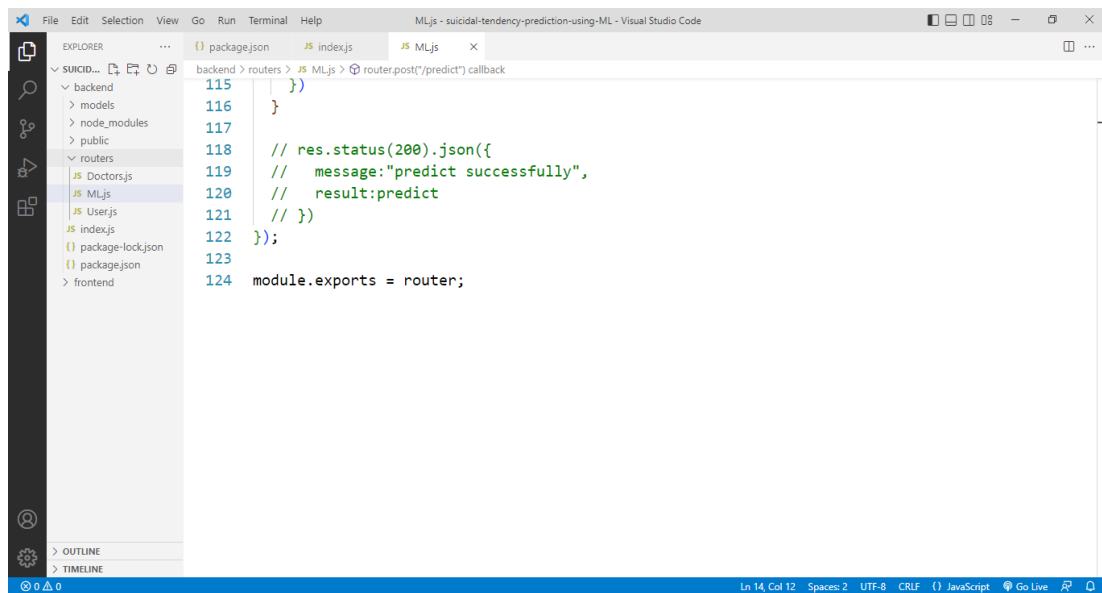


Fig 16 : Backend Code 4

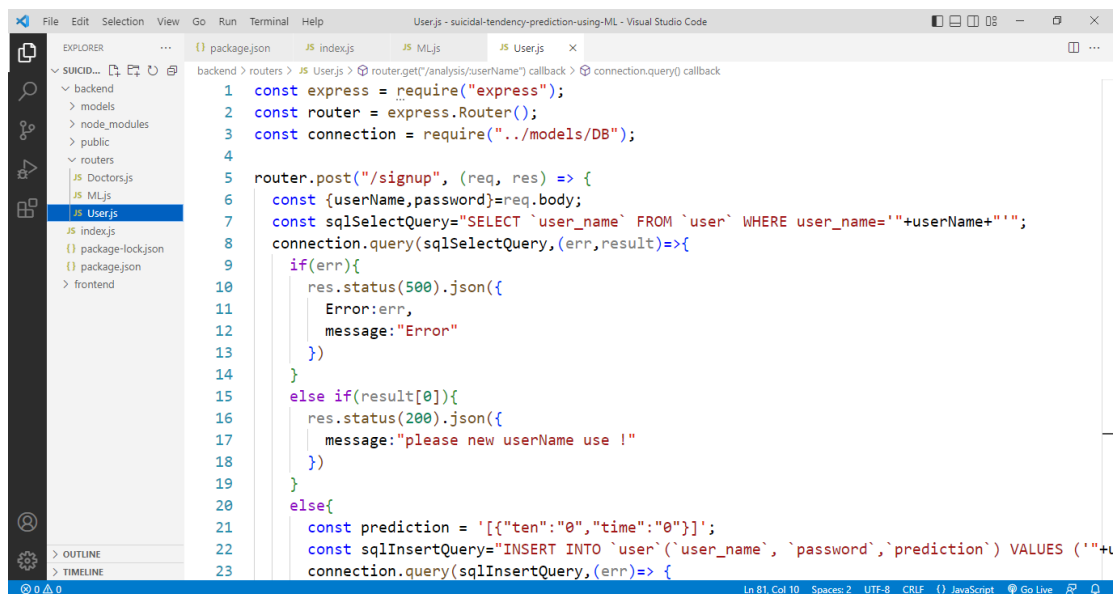
The `module.exports` is a special object which is included in every JavaScript file in the Node.js application by default.



```
115     })
116   }
117
118   // res.status(200).json({
119   //   message:"predict successfully",
120   //   result:predict
121   // })
122 });
123
124 module.exports = router;
```

Fig 17 : Backend Code 5

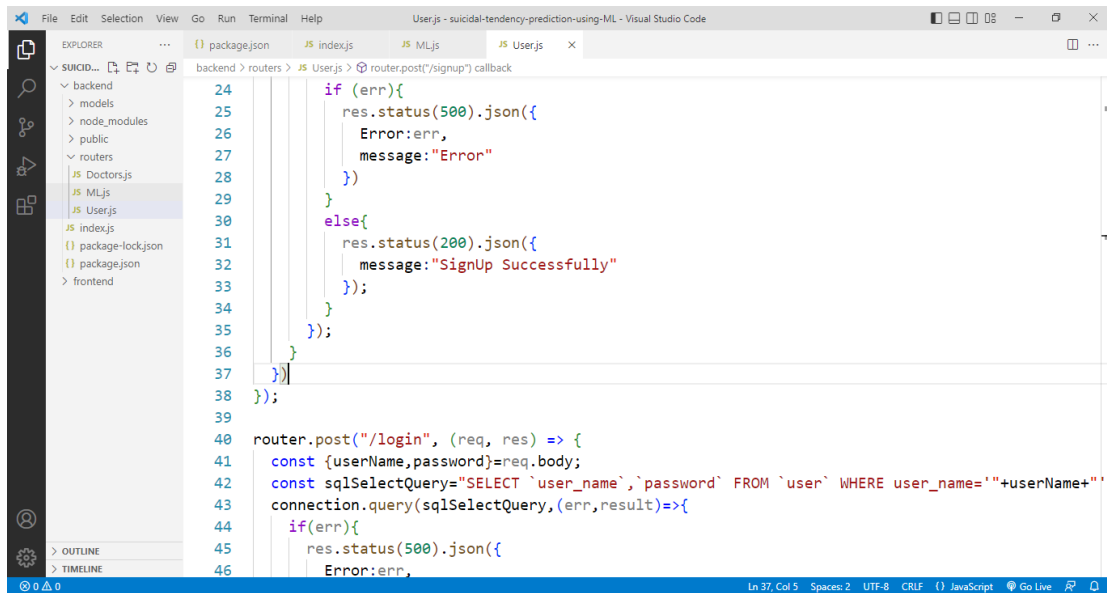
This is backend user router code.



```
1 const express = require("express");
2 const router = express.Router();
3 const connection = require("../models/DB");
4
5 router.post("/signup", (req, res) => {
6   const {userName,password}=req.body;
7   const sqlSelectQuery="SELECT `user_name` FROM `user` WHERE user_name='"+userName+"'";
8   connection.query(sqlSelectQuery,(err,result)=>{
9     if(err){
10       res.status(500).json({
11         Error:err,
12         message:"Error"
13       })
14     }
15     else if(result[0]){
16       res.status(200).json({
17         message:"please new userName use !"
18       })
19     }
20     else{
21       const prediction = '{"ten":0,"time":0}';
22       const sqlInsertQuery="INSERT INTO `user` (`user_name`, `password`, `prediction`) VALUES ('"+
23       connection.query(sqlInsertQuery,(err)=> {
```

Fig 18 : Backend Code 6

This is backend user login router code. In this code frontend request accept and valid user check. If user name and password match then login



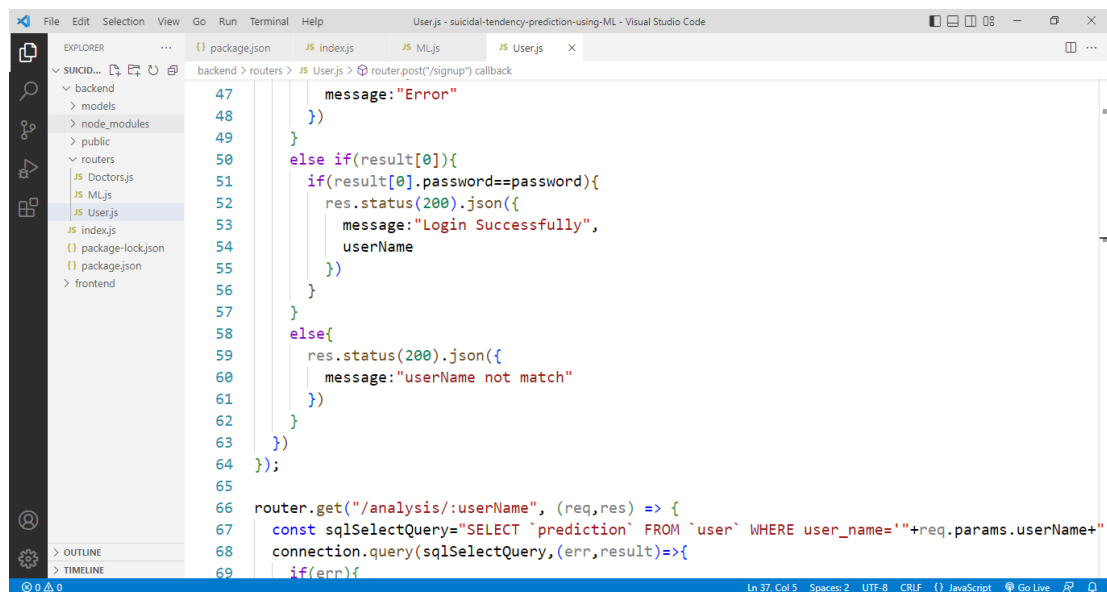
```

24     if (err){
25       res.status(500).json({
26         Error:err,
27         message:"Error"
28       })
29     }
30     else{
31       res.status(200).json({
32         message:"SignUp Successfully"
33       });
34     }
35   });
36 }
37 }
38 });
39
40 router.post("/login", (req, res) => {
41   const {userName,password}=req.body;
42   const sqlSelectQuery="SELECT `user_name`,`password` FROM `user` WHERE user_name='"+userName+"'";
43   connection.query(sqlSelectQuery,(err,result)=>{
44     if(err){
45       res.status(500).json({
46         Error:err,

```

Fig 19 : Backend Code 7

This code check password.



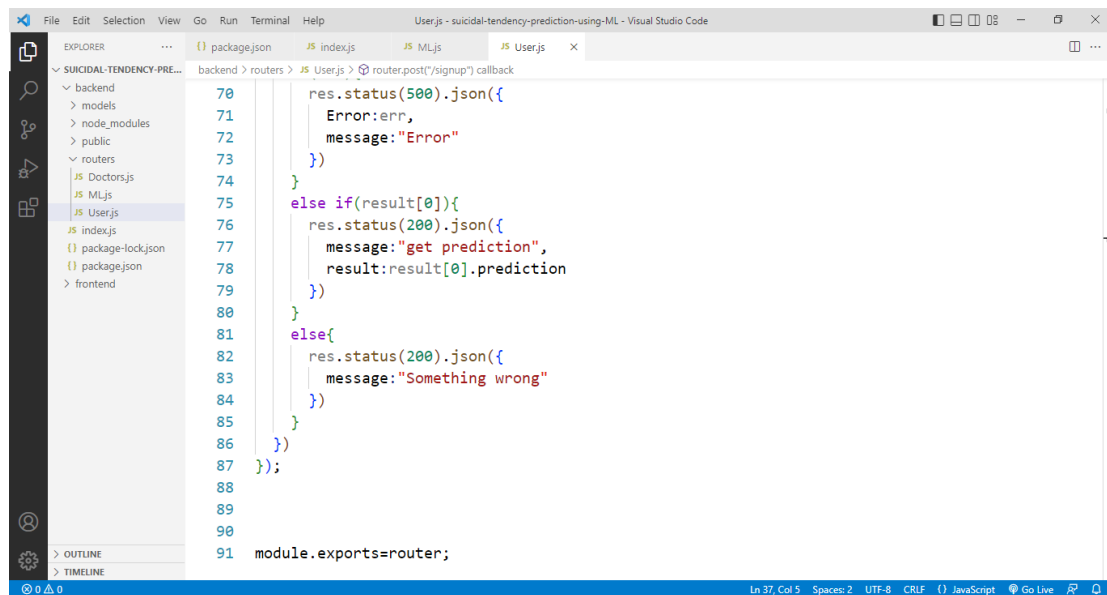
```

47     message:"Error"
48   })
49 }
50 else if(result[0]){
51   if(result[0].password==password){
52     res.status(200).json({
53       message:"Login Successfully",
54       userName
55     })
56   }
57 }
58 else{
59   res.status(200).json({
60     message:"userName not match"
61   })
62 }
63 }
64 });
65
66 router.get("/analysis/:userName", (req,res) => {
67   const sqlSelectQuery="SELECT `prediction` FROM `user` WHERE user_name='"+req.params.userName+"'";
68   connection.query(sqlSelectQuery,(err,result)=>{
69     if(err){

```

Fig 20 : Backend Code 8

This code is backend analysis router. In this code frontend request accept and analysis.

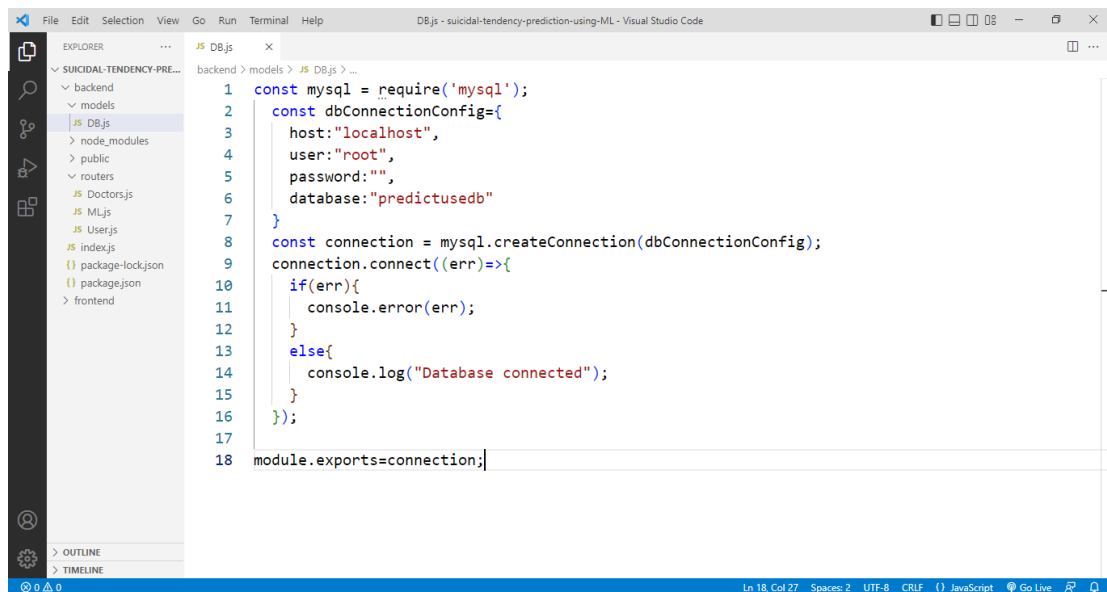


The screenshot shows the Visual Studio Code editor with the file explorer on the left. The project structure includes a 'backend' directory with subdirectories 'models', 'node_modules', 'public', and 'routers'. The 'routers' directory contains 'Doctors.js', 'ML.js', 'User.js', and 'index.js'. The 'User.js' file is selected, and the code editor displays the following JavaScript code:

```
70 res.status(500).json({
71   Error:err,
72   message:"Error"
73 })
74 }
75 else if(result[0]){
76   res.status(200).json({
77     message:"get prediction",
78     result:result[0].prediction
79   })
80 }
81 else{
82   res.status(200).json({
83     message:"Something wrong"
84   })
85 }
86 })
87 });
88
89
90
91 module.exports=router;
```

Fig 21 : Backend Code 9

This is backend database connection code.



The screenshot shows the Visual Studio Code editor with the file explorer on the left. The project structure is the same as in Fig 21. The 'DB.js' file is selected, and the code editor displays the following JavaScript code:

```
1 const mysql = require('mysql');
2 const dbConnectionConfig={
3   host:"localhost",
4   user:"root",
5   password:"",
6   database:"predictusedb"
7 }
8 const connection = mysql.createConnection(dbConnectionConfig);
9 connection.connect((err)=>{
10   if(err){
11     console.error(err);
12   }
13   else{
14     console.log("Database connected");
15   }
16 });
17
18 module.exports=connection;
```

Fig 22 : Backend Code 10

4.14 Backend Deploy

At fast node.js folders and file convert to zip > c-panel > file manager > public_html/userbackendapi upload > convert zip to unzip. This backend produce api.

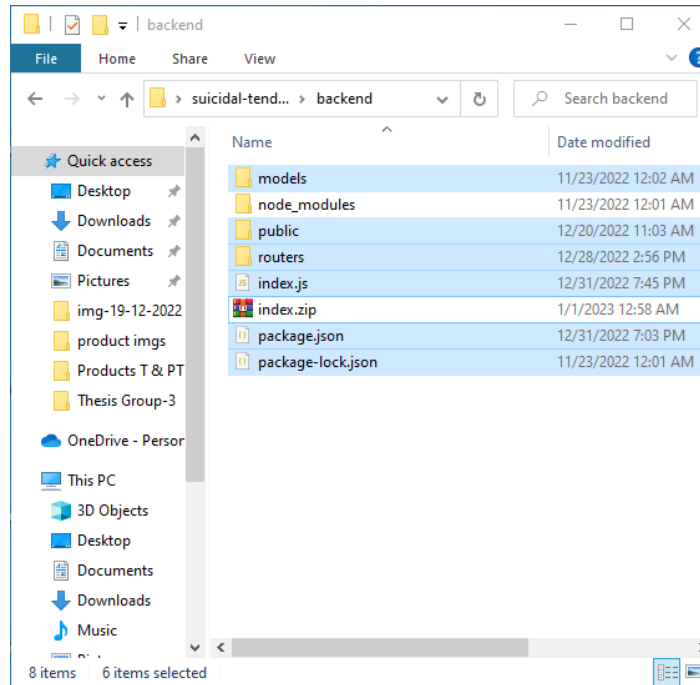


Fig 23 : Backend Files

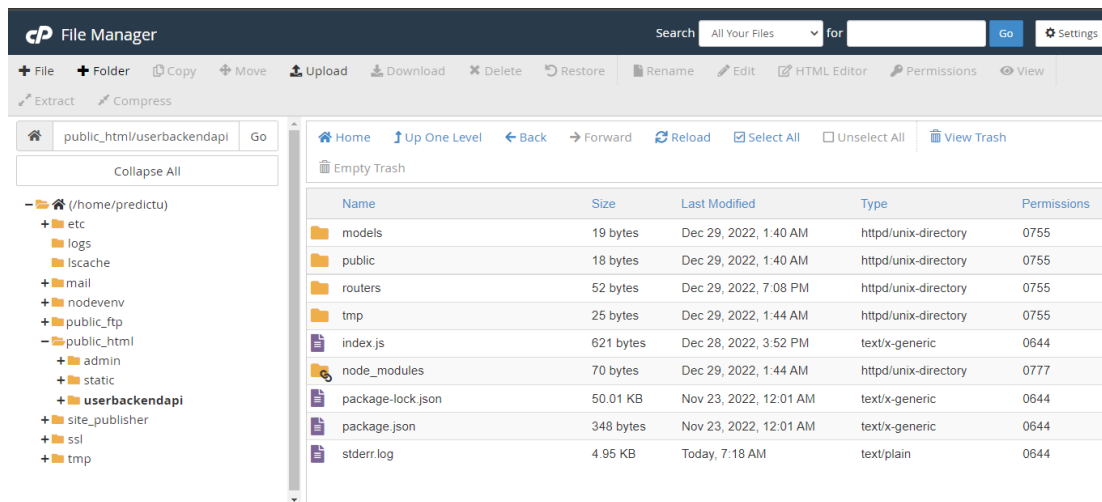


Fig 24 : Backend c-panel Files

This is node.js code upload process at c-panel. Node.js folder path set and package file run then start.

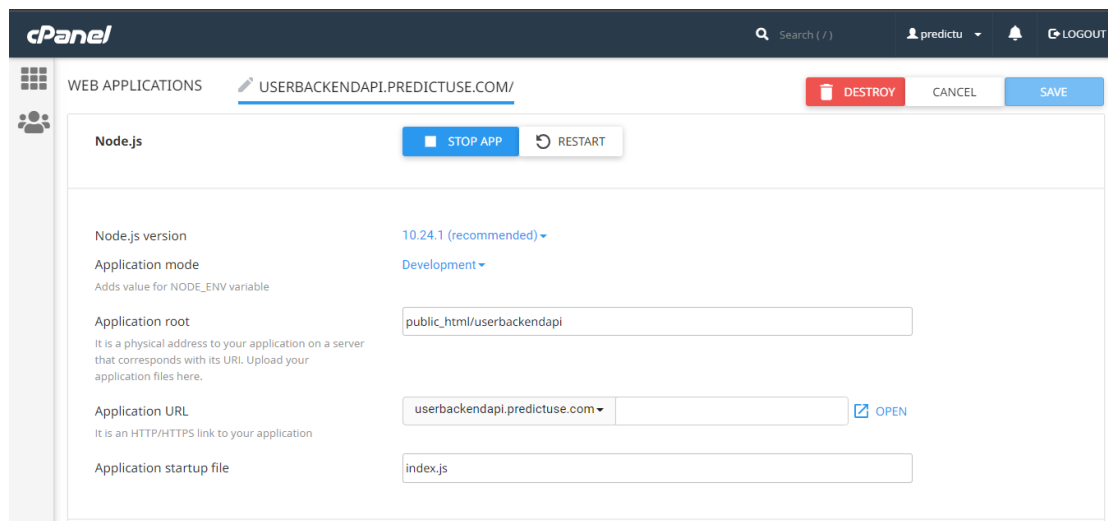


Fig 25 : c-panel nodejs1

In this all package install of project. And environment variable set.

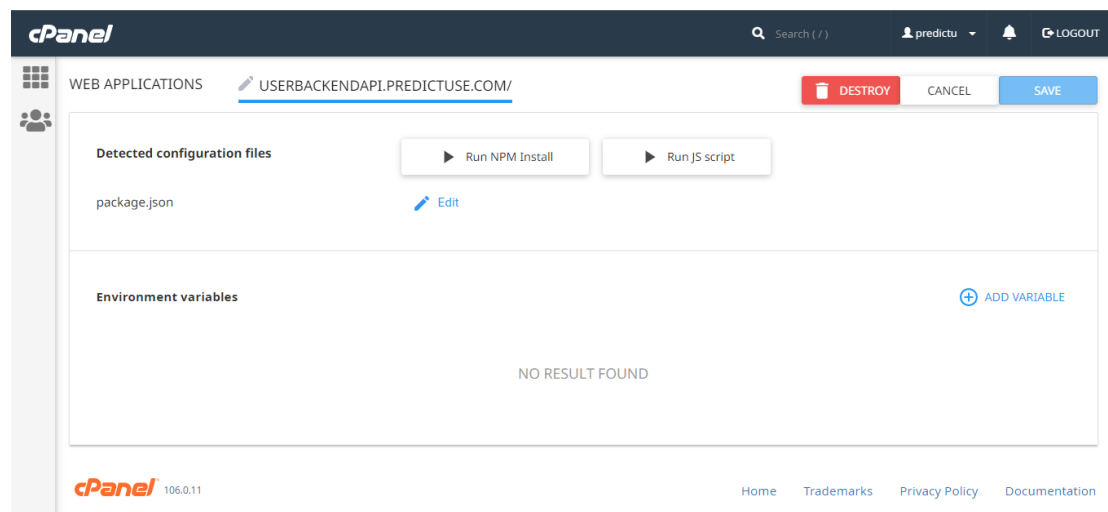


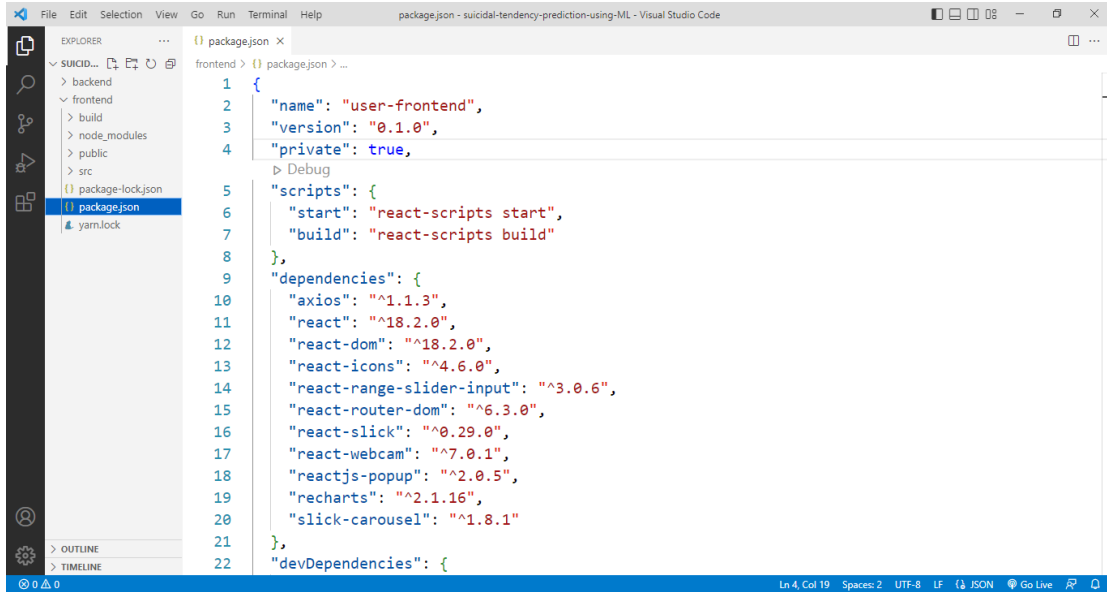
Fig 26 : c-panel nodejs2

4.15 Frontend

- Markup and web languages such as HTML, CSS, JavaScript, and ancillary libraries commonly used in those languages such as Sass or jQuery
- Asynchronous request handling and AJAX
- Single-page applications (with frameworks like React, Angular or Vue.js)
- Web performance (largest contentful paint, time to interactive, 60 FPS animations and interactions, memory usage, etc.)
- Responsive web design
- Cross-browser compatibility issues and workarounds
- End-to-end testing with a headless browser
- Build automation to transform and bundle JavaScript files, reduce image sizes and other processes using tools such as Webpack and Gulp.js
- Search engine optimization
- Accessibility concerns
- Basic usage of image editing tools such as GIMP or Photoshop
- User Interface

4.16 Frontend Code

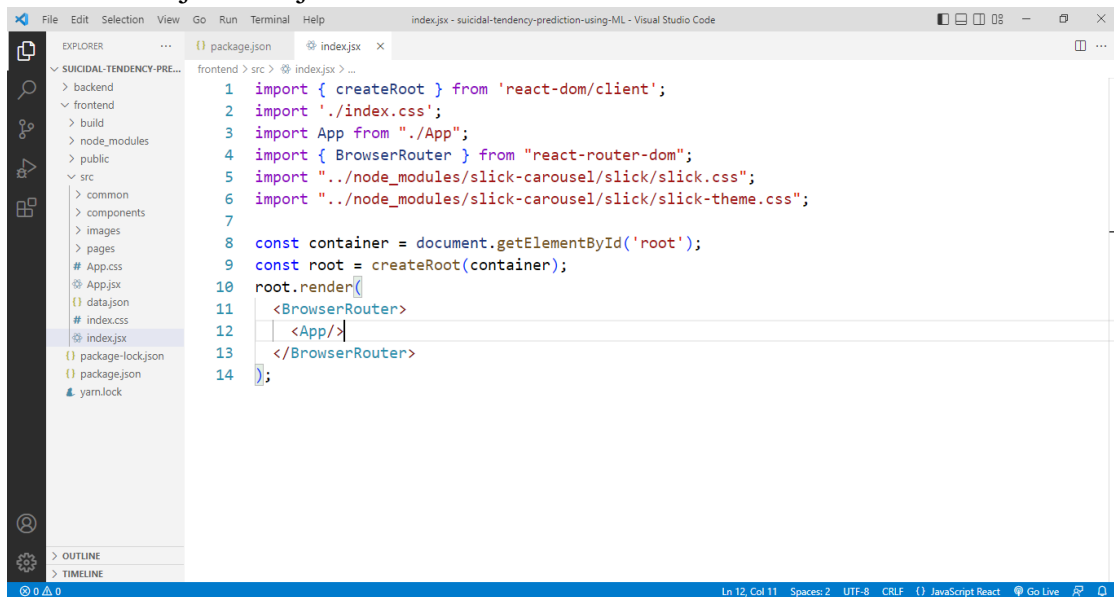
It is React.js json file. This file includes metadata such as name, author and version as well as starter scripts and dependencies used in the project.



```
1 {
2   "name": "user-frontend",
3   "version": "0.1.0",
4   "private": true,
5   "scripts": {
6     "start": "react-scripts start",
7     "build": "react-scripts build"
8   },
9   "dependencies": {
10    "axios": "^1.1.3",
11    "react": "^18.2.0",
12    "react-dom": "^18.2.0",
13    "react-icons": "^4.6.0",
14    "react-range-slider-input": "^3.0.6",
15    "react-router-dom": "^6.3.0",
16    "react-slick": "^0.29.0",
17    "react-webcam": "^7.0.1",
18    "reactjs-popup": "^2.0.5",
19    "recharts": "^2.1.16",
20    "slick-carousel": "^1.8.1"
21  },
22  "devDependencies": {
```

Fig 27 : Frontend code 1

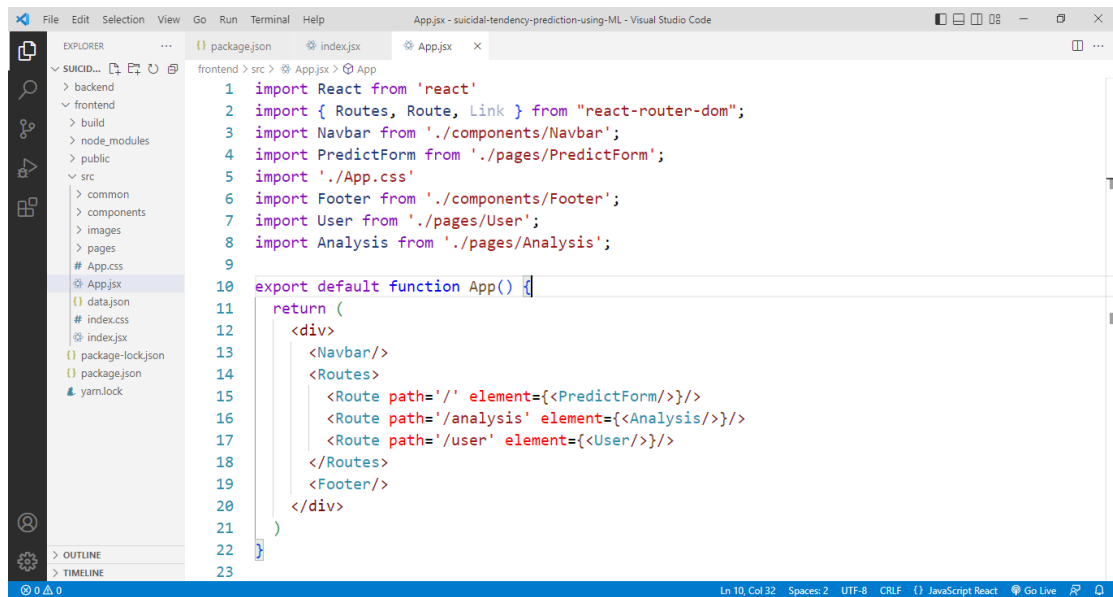
This is React.js index.js code.



```
1 import { createRoot } from 'react-dom/client';
2 import './index.css';
3 import App from './App';
4 import { BrowserRouter } from 'react-router-dom';
5 import './node_modules/slick-carousel/slick/slick.css';
6 import './node_modules/slick-carousel/slick/slick-theme.css';
7
8 const container = document.getElementById('root');
9 const root = createRoot(container);
10 root.render(
11   <BrowserRouter>
12     <App/>
13   </BrowserRouter>
14 );
```

Fig 28 : Frontend code 2

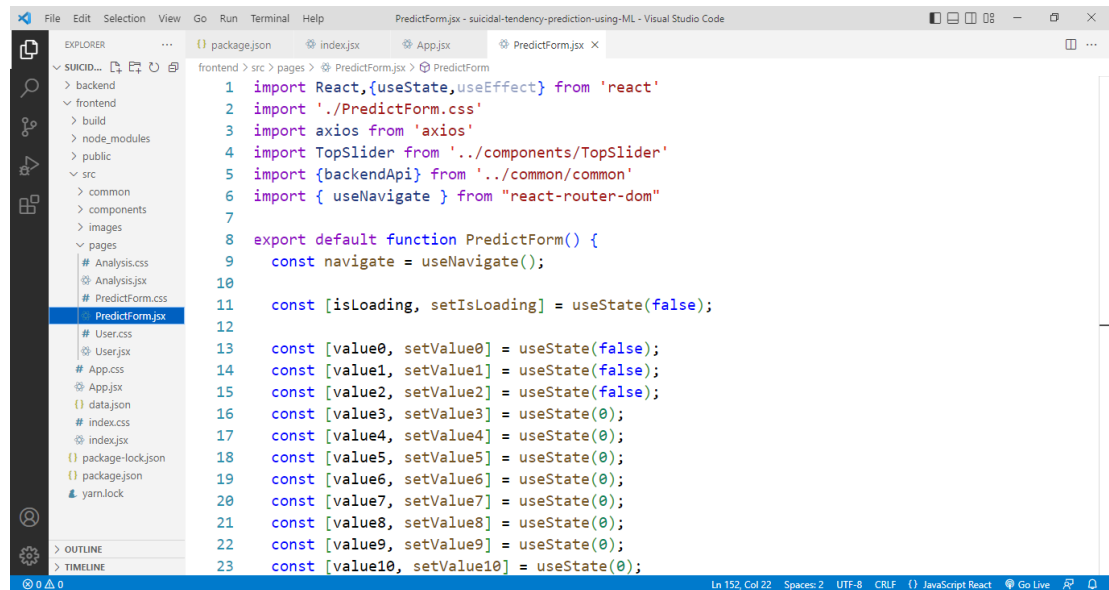
This is App.js file code. In this code all react js file manage and is section manage routing.



```
1 import React from 'react'
2 import { Routes, Route, Link } from "react-router-dom";
3 import Navbar from './components/Navbar';
4 import PredictForm from './pages/PredictForm';
5 import './App.css'
6 import Footer from './components/Footer';
7 import User from './pages/User';
8 import Analysis from './pages/Analysis';
9
10 export default function App() {
11   return (
12     <div>
13       <Navbar/>
14       <Routes>
15         <Route path="/" element={<PredictForm/>}/>
16         <Route path="/analysis" element={<Analysis/>}/>
17         <Route path="/user" element={<User/>}/>
18       </Routes>
19       <Footer/>
20     </div>
21   )
22 }
23
```

Fig 29 : Frontend code 3

This react file is predict input data ui design and request backend with predict data.

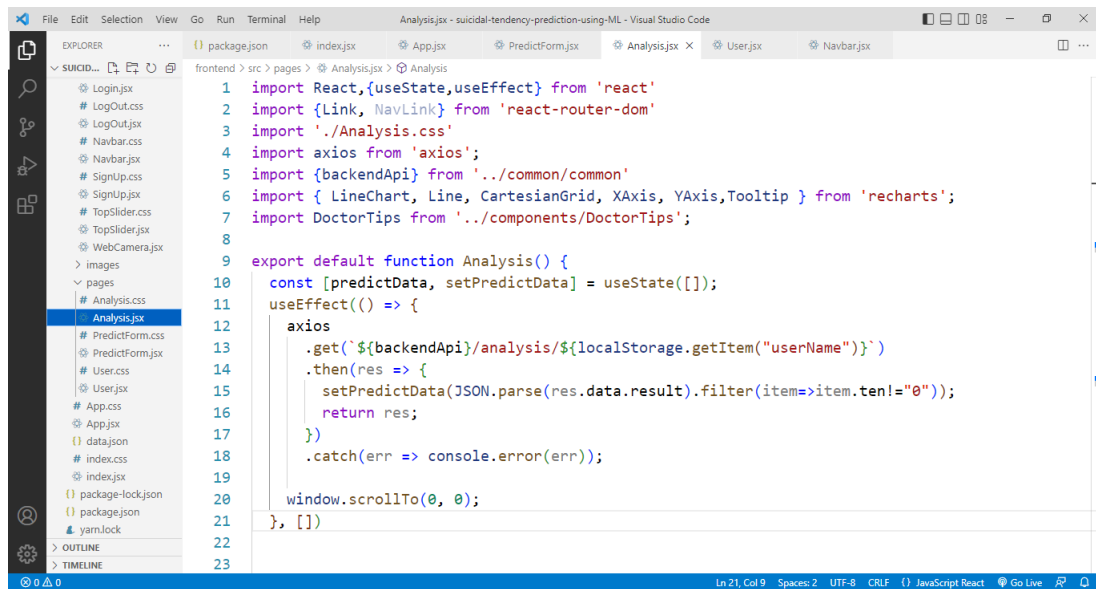


```
1 import React,{useState,useEffect} from 'react'
2 import './PredictForm.css'
3 import axios from 'axios'
4 import TopSlider from './components/TopSlider'
5 import {backendApi} from '../common/common'
6 import { useNavigate } from "react-router-dom"
7
8 export default function PredictForm() {
9   const navigate = useNavigate();
10
11   const [isLoading, setIsLoading] = useState(false);
12
13   const [value0, setValue0] = useState(false);
14   const [value1, setValue1] = useState(false);
15   const [value2, setValue2] = useState(false);
16   const [value3, setValue3] = useState(0);
17   const [value4, setValue4] = useState(0);
18   const [value5, setValue5] = useState(0);
19   const [value6, setValue6] = useState(0);
20   const [value7, setValue7] = useState(0);
21   const [value8, setValue8] = useState(0);
22   const [value9, setValue9] = useState(0);
23   const [value10, setValue10] = useState(0);

```

Fig 30 : Frontend code 4

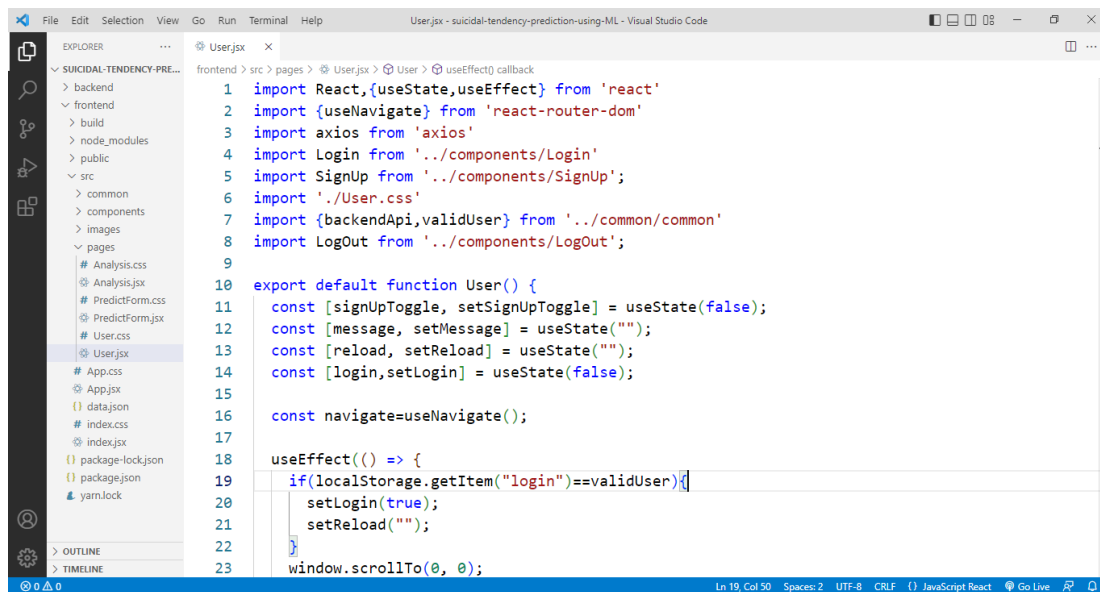
In this react file analysis part and analysis ui desing. This code accept backend data and show chart.



```
1 import React,{useState,useEffect} from 'react'
2 import {Link, NavLink} from 'react-router-dom'
3 import './Analysis.css'
4 import axios from 'axios';
5 import {backendApi} from '../common/common'
6 import { LineChart, Line, CartesianGrid, XAxis, YAxis,Tooltip } from 'recharts';
7 import DoctorTips from '../components/DoctorTips';
8
9 export default function Analysis() {
10   const [predictData, setPredictData] = useState([]);
11   useEffect(() => {
12     axios
13       .get(`${backendApi}/analysis/${localStorage.getItem("userName")}`)
14       .then(res => {
15         setPredictData(JSON.parse(res.data.result).filter(item=>item.ten!="0"));
16         return res;
17       })
18       .catch(err => console.error(err));
19   }, []);
20   window.scrollTo(0, 0);
21 }, [])
```

Fig 31 : Frontend code 5

This route user login and signup ui. User data communicate backend.



```
1 import React,{useState,useEffect} from 'react'
2 import {useNavigate} from 'react-router-dom'
3 import axios from 'axios'
4 import Login from '../components/Login'
5 import SignUp from '../components/SignUp';
6 import './User.css'
7 import {backendApi,validUser} from '../common/common'
8 import Logout from '../components/Logout';
9
10 export default function User() {
11   const [signUpToggle, setSignUpToggle] = useState(false);
12   const [message, setMessage] = useState("");
13   const [reload, setReload] = useState("");
14   const [login,setLogin] = useState(false);
15
16   const navigate=useNavigate();
17
18   useEffect(() => {
19     if(localStorage.getItem("login")==validUser){
20       setLogin(true);
21       setReload("");
22     }
23     window.scrollTo(0, 0);
24   }, [])
```

Fig 32 : Frontend code 6

4.17 Frontend Deploy

At fast React Code, Build > zip > c-panel > public_html folder > upload > unzip

SANJOY ROY@DESKTOP-6L8729J MINGW64 ~/Desktop/suicidal-tendency-prediction-using-ML/frontend

\$ npm run build

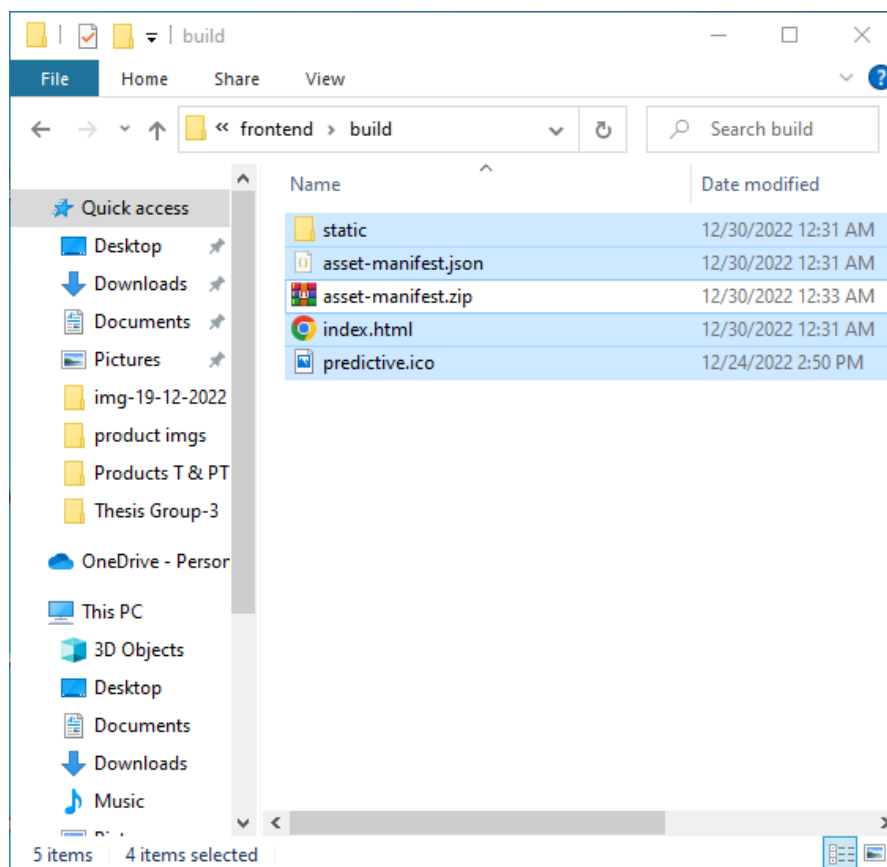


Fig 33 : Frontend Files

4.18 Prediction

Predict 1

```
const handlePredict=()=>=>{
  if(confirm("predict")){
    setIsLoading(true);
    axios
      .post(`${backendApi}/predict`,(localStorage.getItem('login')==validUser)?
        {...totalValue,login:'true',userName:localStorage.getItem('userName')}:
        {...totalValue,login:'false',userName:null})
      .then(res =>{
        const predictSingle=[ {
          ten:res.data.result.tendency,
          time:new Date().toLocaleDateString()
        } ]
        localStorage.setItem("predictSingle",JSON.stringify(predictSingle));
        navigate("/analysis");
        setIsLoading(false);
        window.scrollTo(0, 0);
        return res;
      })
      .catch(err => console.error(err));
  }
}
```

The image shows a user interface for data input. It features two horizontal sliders. The first slider is labeled 'Social isolation ?' and 'সামাজিক বিচ্ছিন্নতা ?' with a value of 55%. The second slider is labeled 'Suicide clusters in communities ?' and 'সম্প্রদায়ের মধ্যে আত্মহত্যার ক্লাস্টার ?' with a value of 50%. Both sliders have a scale from 0 to 100. Below the sliders is a blue button labeled 'Predict'.

Fig 34 : Data Input 1

Data Training

```
const net=new brain.NeuralNetwork();

net.train([

  { input: { mValue1: 100, mValue2: 100, mValue2: 100,mValue2: 100, mValue5:
100, mValue6: 100, mValue7: 100,mValue8: 100, mValue9: 100, mValue10: 100,
mValue11: 100, mValue12: 100, mValue13: 100, mValue14: 100, mValue15: 100
}, output: { tendency: 1 } },

  { input: { mValue1: 50, mValue2: 50, mValue2: 50,mValue2: 50, mValue5: 50,
mValue6: 50, mValue7: 50,mValue8: 50, mValue9: 50, mValue10: 50, mValue11:
50, mValue12: 50, mValue13: 50, mValue14: 50, mValue15: 50 }, output: {
tendency: 1.5 } },

  { input: { mValue1: 0, mValue2: 0, mValue2: 0,mValue2: 0, mValue5: 0,
mValue6: 0, mValue7: 0,mValue8: 0, mValue9: 0, mValue10: 0, mValue11: 0,
mValue12: 0, mValue13: 0, mValue14: 0, mValue15: 0 }, output: { tendency: 0 }
},

]);

let prediction=net.run(bodyData);
```

Analysis

```
useEffect(() => {  
  
  axios  
    .get(`${backendApi}/analysis/${localStorage.getItem("userName")}`)  
    .then(res => {  
      setPredictData(JSON.parse(res.data.result).filter(item=>item.ten!="0"));  
      return res;  
    })  
    .catch(err => console.error(err));  
  
  window.scrollTo(0, 0);  
}, [])
```

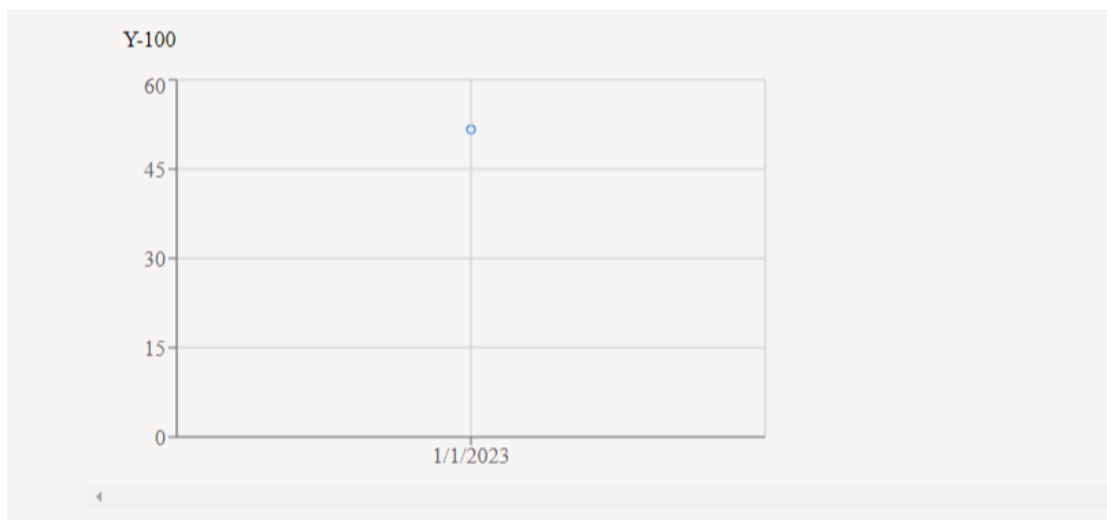


Fig 35 : Predict Result 1

Predict 2

```
const handlePredict=()=>=>{
  if(confirm("predict")){
    setIsLoading(true);
    axios
      .post(`${backendApi}/predict`,(localStorage.getItem('login')==validUser)?
        {...totalValue,login:'true',userName:localStorage.getItem('userName')}:
        {...totalValue,login:'false',userName:null})
      .then(res =>{
        const predictSingle=[{
          ten:res.data.result.tendency,
          time:new Date().toLocaleDateString()
        }]
        localStorage.setItem("predictSingle",JSON.stringify(predictSingle));
        navigate("/analysis");
        setIsLoading(false);
        window.scrollTo(0, 0);
        return res;
      })
      .catch(err => console.error(err));
  }
}
```

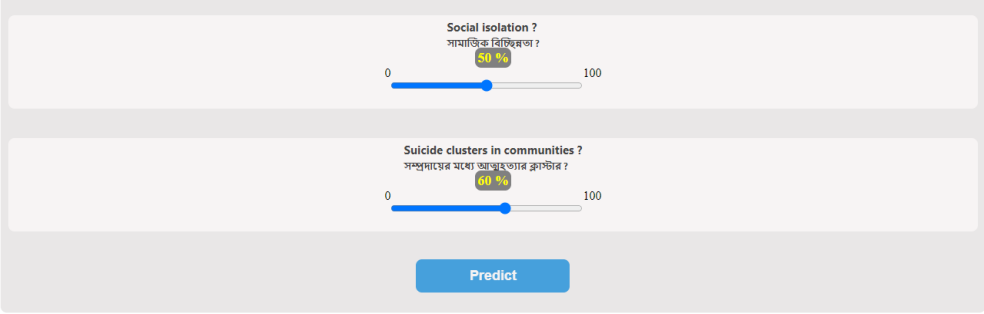


Fig 36 : Data Input 2

Data Training

```
const net=new brain.NeuralNetwork();

net.train([

  { input: { mValue1: 100, mValue2: 100, mValue2: 100,mValue2: 100, mValue5:
100, mValue6: 100, mValue7: 100,mValue8: 100, mValue9: 100, mValue10: 100,
mValue11: 100, mValue12: 100, mValue13: 100, mValue14: 100, mValue15: 100
}, output: { tendency: 1 } },

  { input: { mValue1: 50, mValue2: 50, mValue2: 50,mValue2: 50, mValue5: 50,
mValue6: 50, mValue7: 50,mValue8: 50, mValue9: 50, mValue10: 50, mValue11:
50, mValue12: 50, mValue13: 50, mValue14: 50, mValue15: 50 }, output: {
tendency: 1.5 } },

  { input: { mValue1: 0, mValue2: 0, mValue2: 0,mValue2: 0, mValue5: 0,
mValue6: 0, mValue7: 0,mValue8: 0, mValue9: 0, mValue10: 0, mValue11: 0,
mValue12: 0, mValue13: 0, mValue14: 0, mValue15: 0 }, output: { tendency: 0 }
},

]);

let prediction=net.run(bodyData);
```

Analysis

```
useEffect(() => {  
  
  axios  
    .get(`${backendApi}/analysis/${localStorage.getItem("userName")}`)  
    .then(res => {  
      setPredictData(JSON.parse(res.data.result).filter(item=>item.ten!="0"));  
      return res;  
    })  
    .catch(err => console.error(err));  
  
  window.scrollTo(0, 0);  
}, [])
```

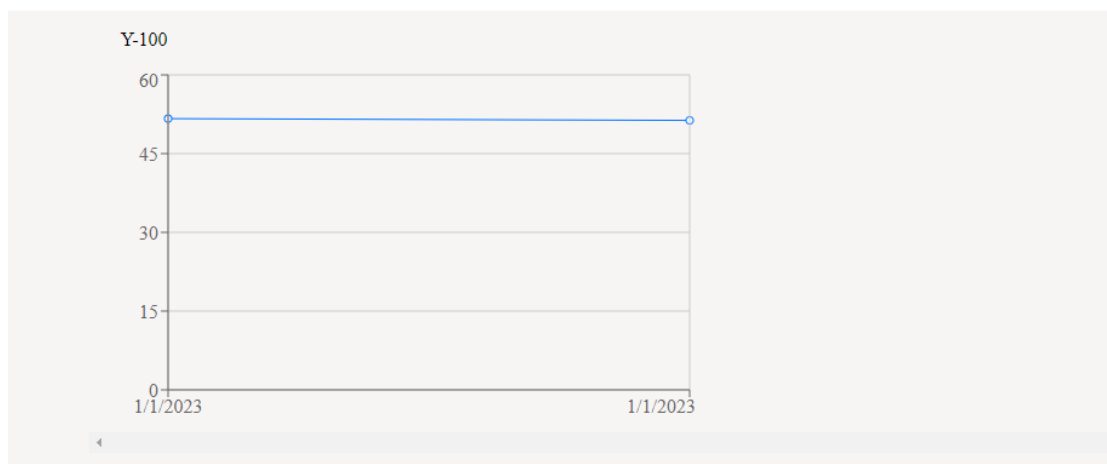


Fig 37 : Predict Result 2

Predict 3

```
const handlePredict=()=>=>{
  if(confirm("predict")){
    setIsLoading(true);
    axios
      .post(`${backendApi}/predict`,(localStorage.getItem('login')==validUser)?
        {...totalValue,login:'true',userName:localStorage.getItem('userName')}:
        {...totalValue,login:'false',userName:null})
      .then(res =>{
        const predictSingle=[{
          ten:res.data.result.tendency,
          time:new Date().toLocaleDateString()
        }]
        localStorage.setItem("predictSingle",JSON.stringify(predictSingle));
        navigate("/analysis");
        setIsLoading(false);
        window.scrollTo(0, 0);
        return res;
      })
      .catch(err => console.error(err));
  }
}
```

The image shows a user interface for data input. It features two horizontal sliders. The first slider is labeled 'Social isolation ?' with the Bengali text 'সামাজিক বিচ্ছিন্নতা ?' below it; the slider is set to 45%. The second slider is labeled 'Suicide clusters in communities ?' with the Bengali text 'সম্প্রদায়ের মধ্যে আত্মহত্যার ক্লাস্টার ?' below it; the slider is set to 50%. Both sliders have a scale from 0 to 100. At the bottom of the form is a blue button labeled 'Predict'.

Fig 38 : Data Input 3

Data Training

```
const net=new brain.NeuralNetwork();

net.train([

  { input: { mValue1: 100, mValue2: 100, mValue2: 100,mValue2: 100, mValue5:
100, mValue6: 100, mValue7: 100,mValue8: 100, mValue9: 100, mValue10: 100,
mValue11: 100, mValue12: 100, mValue13: 100, mValue14: 100, mValue15: 100
}, output: { tendency: 1 } },

  { input: { mValue1: 50, mValue2: 50, mValue2: 50,mValue2: 50, mValue5: 50,
mValue6: 50, mValue7: 50,mValue8: 50, mValue9: 50, mValue10: 50, mValue11:
50, mValue12: 50, mValue13: 50, mValue14: 50, mValue15: 50 }, output: {
tendency: 1.5 } },

  { input: { mValue1: 0, mValue2: 0, mValue2: 0,mValue2: 0, mValue5: 0,
mValue6: 0, mValue7: 0,mValue8: 0, mValue9: 0, mValue10: 0, mValue11: 0,
mValue12: 0, mValue13: 0, mValue14: 0, mValue15: 0 }, output: { tendency: 0 }
},

]);

let prediction=net.run(bodyData);
```

Analysis

```
useEffect(() => {  
  
  axios  
    .get(`${backendApi}/analysis/${localStorage.getItem("userName")}`)  
    .then(res => {  
      setPredictData(JSON.parse(res.data.result).filter(item=>item.ten!="0"));  
      return res;  
    })  
    .catch(err => console.error(err));  
  
  window.scrollTo(0, 0);  
}, [])
```

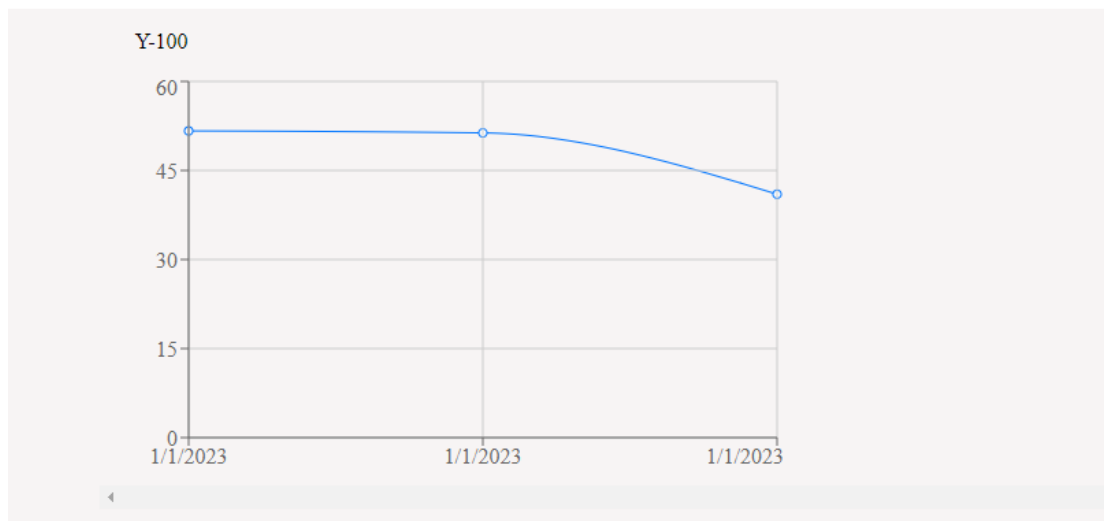
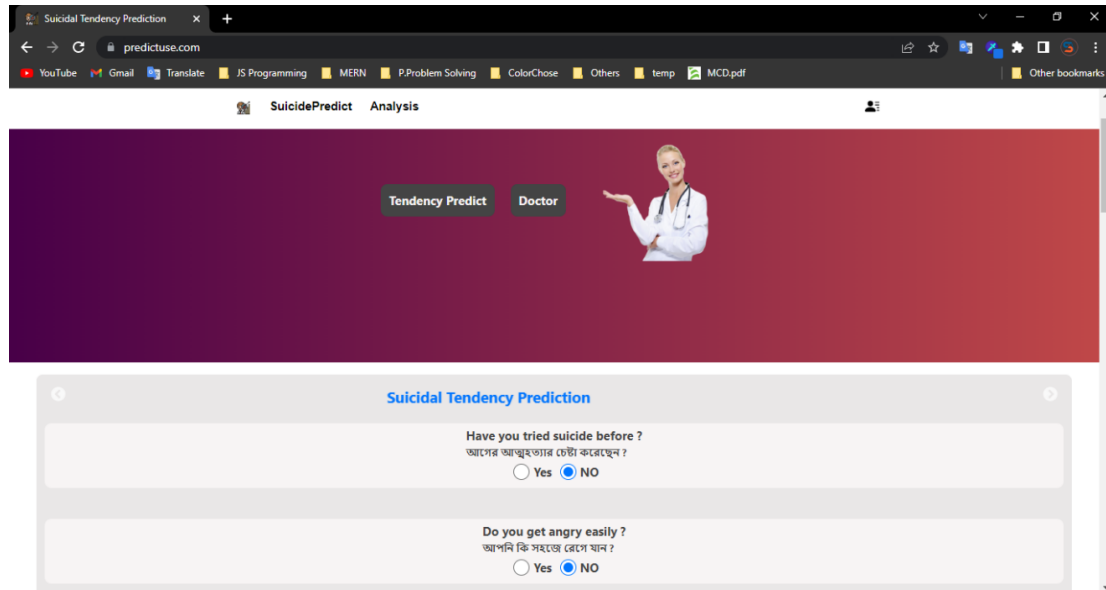


Fig 39 : Predict Result 3

4.19 Finally Complete Our Siucidal Tendency Prediction Web Site

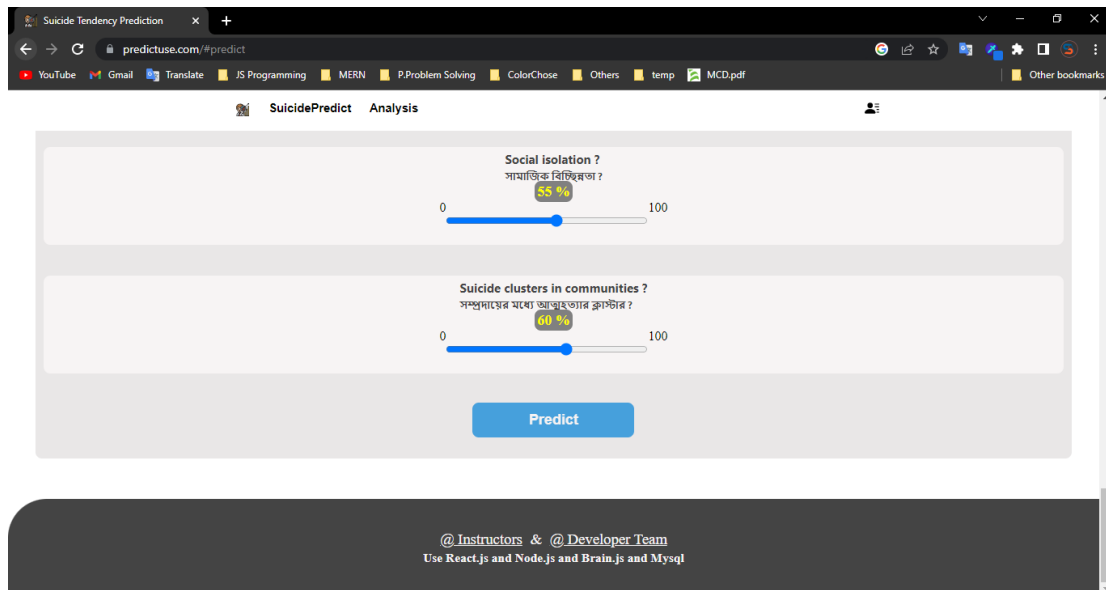


The screenshot shows the home page of the 'Suicide Tendency Prediction' website. The browser address bar shows 'predictuse.com'. The page has a dark purple header with a navigation bar containing 'SuicidePredict' and 'Analysis'. Below the header is a large red banner with a female doctor icon and two buttons: 'Tendency Predict' and 'Doctor'. The main content area is titled 'Suicidal Tendency Prediction' and contains two questions with radio button options:

- Have you tried suicide before ?
আপের আত্মহত্যার চেষ্টা করেছেন ?
☐ Yes ☒ NO
- Do you get angry easily ?
আপনি কি সহজে রেগে যান ?
☐ Yes ☒ NO

Fig 40 : WebSite Home Page

In this question data input and predict.



The screenshot shows the data input page of the 'Suicide Tendency Prediction' website. The browser address bar shows 'predictuse.com/#predict'. The page has a dark purple header with a navigation bar containing 'SuicidePredict' and 'Analysis'. Below the header is a large red banner with a female doctor icon and two buttons: 'Tendency Predict' and 'Doctor'. The main content area is titled 'Suicidal Tendency Prediction' and contains two questions with sliders:

- Social isolation ?
সামাজিক বিচ্ছিন্নতা ?
0 55 % 100
- Suicide clusters in communities ?
সম্প্রদায়ের মধ্যে আত্মহত্যার ক্লাস্টার ?
0 60 % 100

Below the sliders is a blue 'Predict' button. At the bottom of the page, there is a footer with the text: '@Instructors & @Developer Team' and 'Use React.js and Node.js and Braia.js and Mysql'.

Fig 41 : WebSite Data Input Page

This is analysis page. This page show predict result chart. Y axis is 1 to 100 and X axis is auto. Graph line is high suicide tendency high.

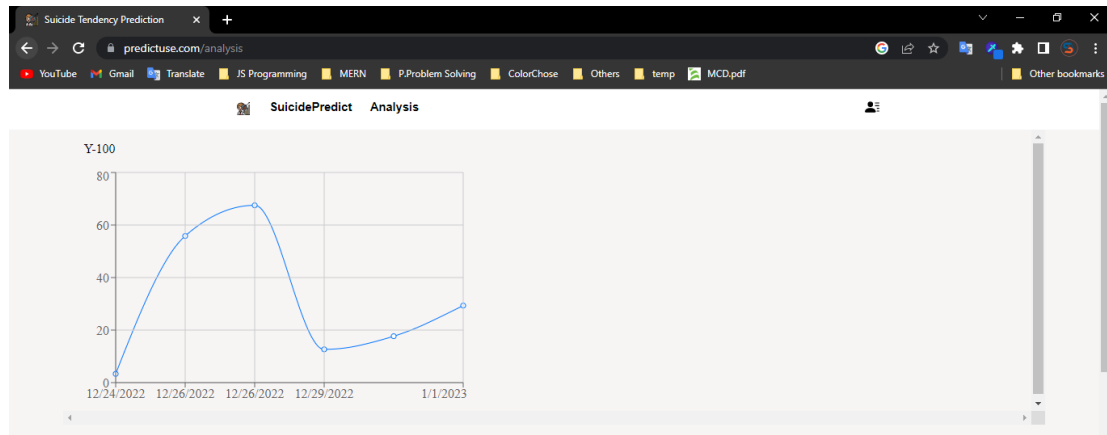


Fig 42 : WebSite Result Page

In this login and signUp page. At first signUp for need user name and password.then login.

User Name

Password

Login

New User SignUp

Fig 43 : WebSite Login and SignUp page

Chapter 5

Result & Conclusions

Result

Prediction Analysis Report analyzes the history of metrics and provide a prediction for the future. A Prediction Analysis Report applies built-in prediction mechanisms and sophisticated forecasting techniques on historical data for a metric to automatically compute how that metric is likely to change in the future. The result is a graph where previous predictions and future predictions are both plotted together.

Suicide tendency is 51.66%.

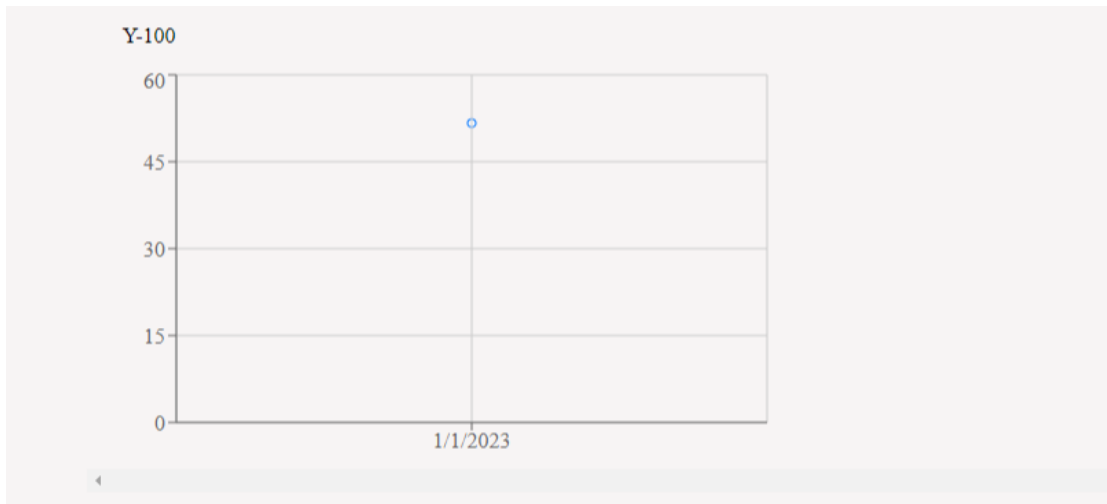


Fig 44 : Result 1

Suicide tendency is 51.33%.

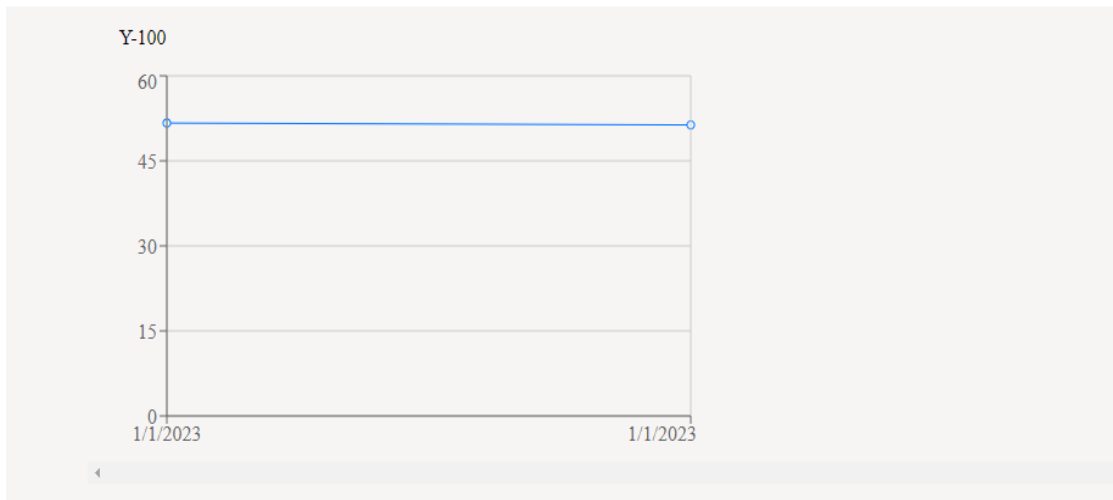


Fig 45 : Result 2

Suicide tendency is 41%.

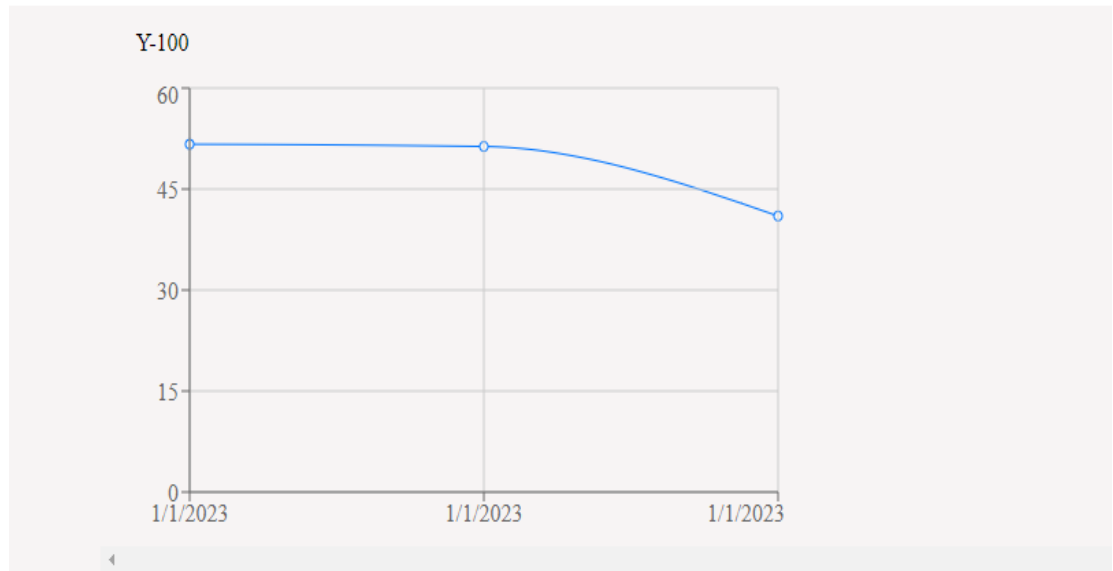


Fig 46 : Result 3

Conclusions

This project work will be really useful and helpful in studying how useful is the machine learning when it come on analyzing the data. There is a really a big difference between studying through books and applying the things learned in the real life. In this project we were able to practically implement our knowledge of machine learning. This project also taught us about various difference to word on to beautiful our project and how can creative thinking lead to such an interesting creation of beautiful things. Finally, this project has a great benefit to us.

Note that, the prediction accuracy is not 100% accurate, cause no system is able to predict 100% result. But if we compare the prediction result with our saved data, we can say that the prediction is impressive and acceptable.

REFERENCES

[1] [HTML] Suicide prediction with machine learning.

G Rakesh - American Journal of Psychiatry Residents', 2017 - Am Psychiatric Assoc. This study utilized machine learning to predict suicide risk. The STARRS model was able to predict suicide risk with of previous suicidal ideation. https://psychiatryonline.org/doi/10.1176/appi.ajprj.2017.120105?utm_source=TrendMD&utm_medium=cpc&utm_campaign=American_Journal_of_Psychiatry_Residents%27_Journal_TrendMD_0

[2] MERN Stack is a collection of powerful technologies and robust, used to develop scalable master web applications comprising backend, front-end, and database. <https://www.javatpoint.com/mern-stack>

[3] GPU accelerated Neural Networks in JavaScript, for Browsers and Node.js. Brain.js performs computations using GPU and gracefully fallback to pure You've visited this page many times. Last visit: 11/5/22 <https://brain.js.org>

[4] suicidal tendency prediction. <https://www.wikipedia.org>

[5] suicidal tendency prediction. <https://scholar.google.com>

[6] Research & Journals <https://eub.edu.bd>