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 your 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 gate proper prediction.

Suppose, your 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**](https://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.

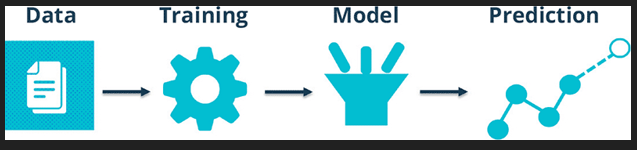


Fig 1 : ML Model

**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.

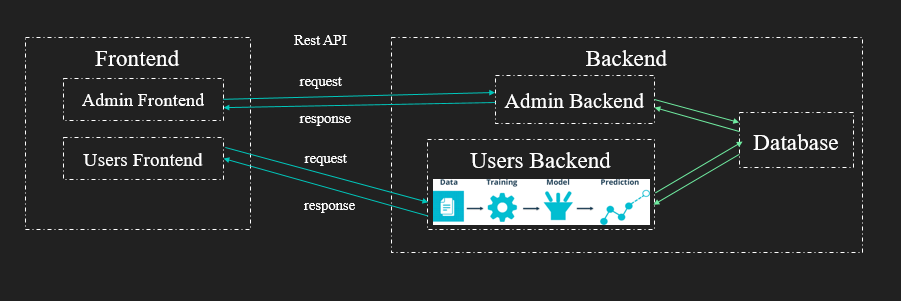


Fig 2 : Project Model

**4.3 Visual Studio Code**

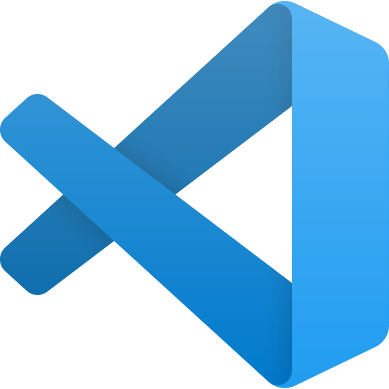
Visual Studio Code, also commonly referred to as VS Code, is a source-code editor made by Microsoft with the Electron Framework, for Windows, Linux and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git.

Fig 3 : Visual Studio Code

**4.4 Node.js**

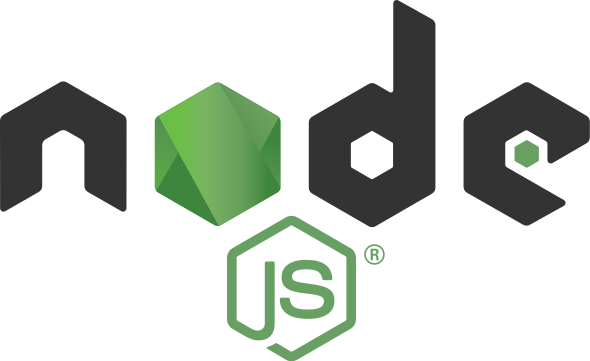
Node.js is an [open-source](https://en.wikipedia.org/wiki/Open-source_software) server environment. Node.js is [cross-platform](https://en.wikipedia.org/wiki/Cross-platform) and runs on Windows, Linux, Unix, and macOS. Node.js is a [back-end](https://en.wikipedia.org/wiki/Front_end_and_back_end) [JavaScript](https://en.wikipedia.org/wiki/JavaScript) [runtime environment](https://en.wikipedia.org/wiki/Runtime_system). Node.js runs on the [V8](https://en.wikipedia.org/wiki/V8_(JavaScript_engine)) [JavaScript Engine](https://en.wikipedia.org/wiki/JavaScript_Engine) and executes JavaScript code outside a [web browser](https://en.wikipedia.org/wiki/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.

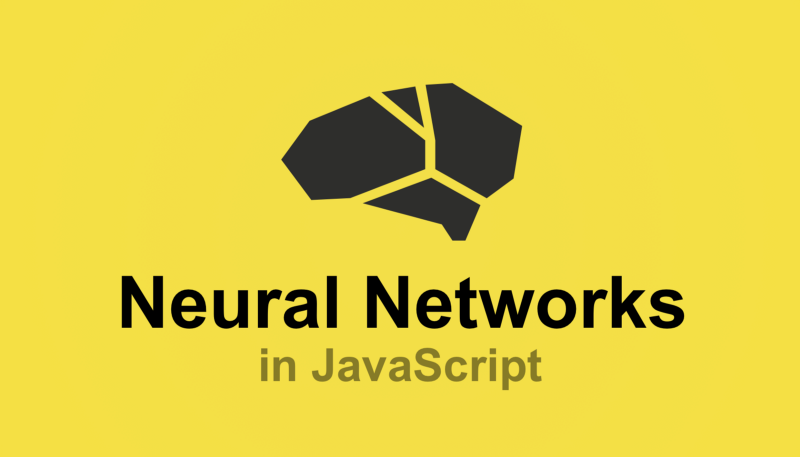


Fig 6 : Brain.js

**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.

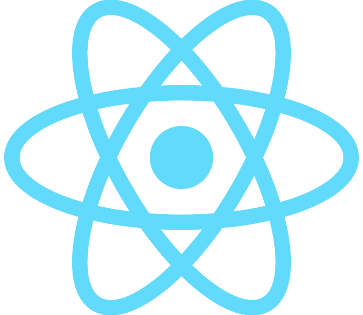


Fig 8 : React.js

**4.9 Axios**

In a nutshell, Axios is a Javascript library used to make HTTP requests from [node.js](https://nodejs.org/en/) or XMLHttpRequests from the browser that also supports the [ES6 Promise API](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Promise). 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**

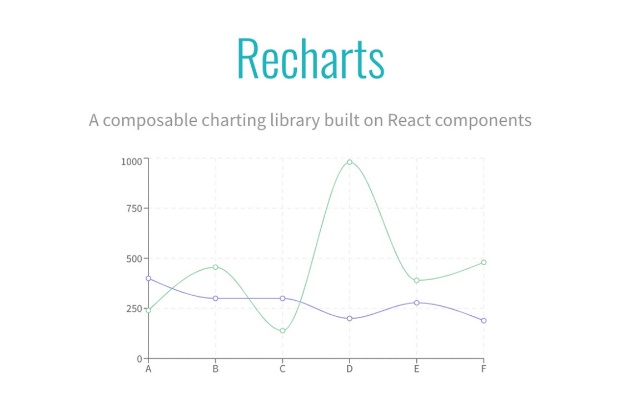
Recharts is a Redefined chart library built with React and D3. The main purpose of this library is to help you to write charts in React applications without any pain. Main principles of Recharts are: Simply deploy with React components. Native SVG support, lightweight depending only on some D3 submodules

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.

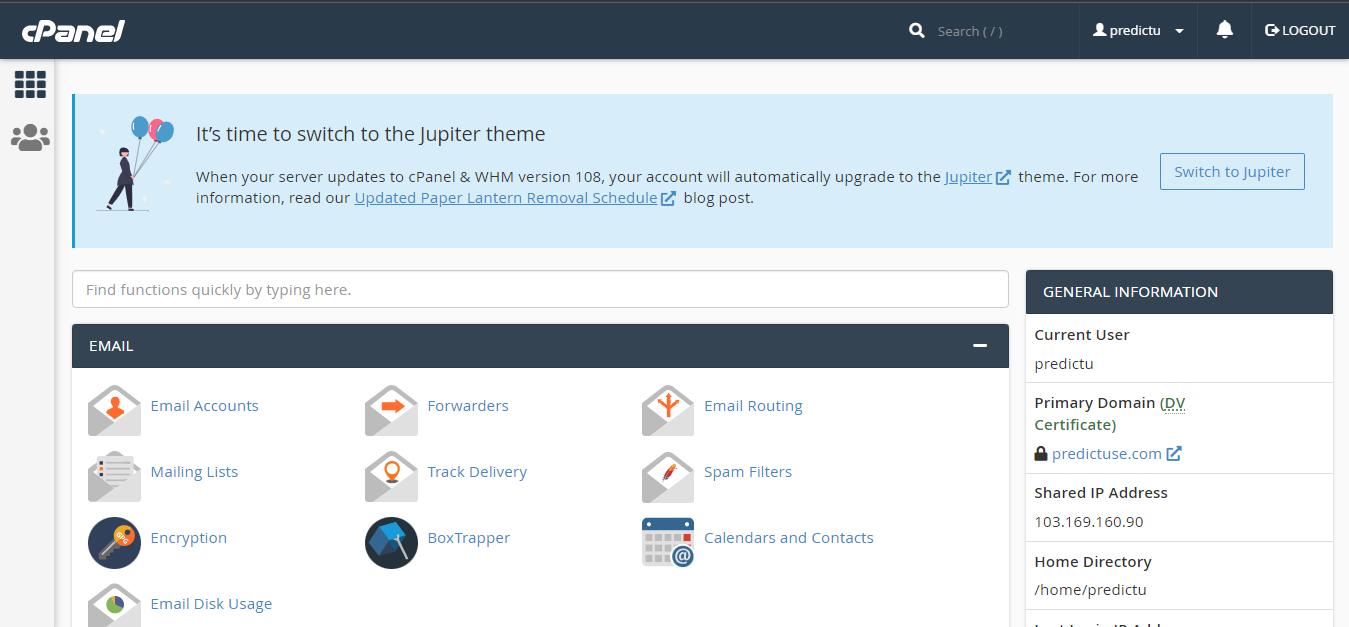


Fig 11 : c-pannel 1

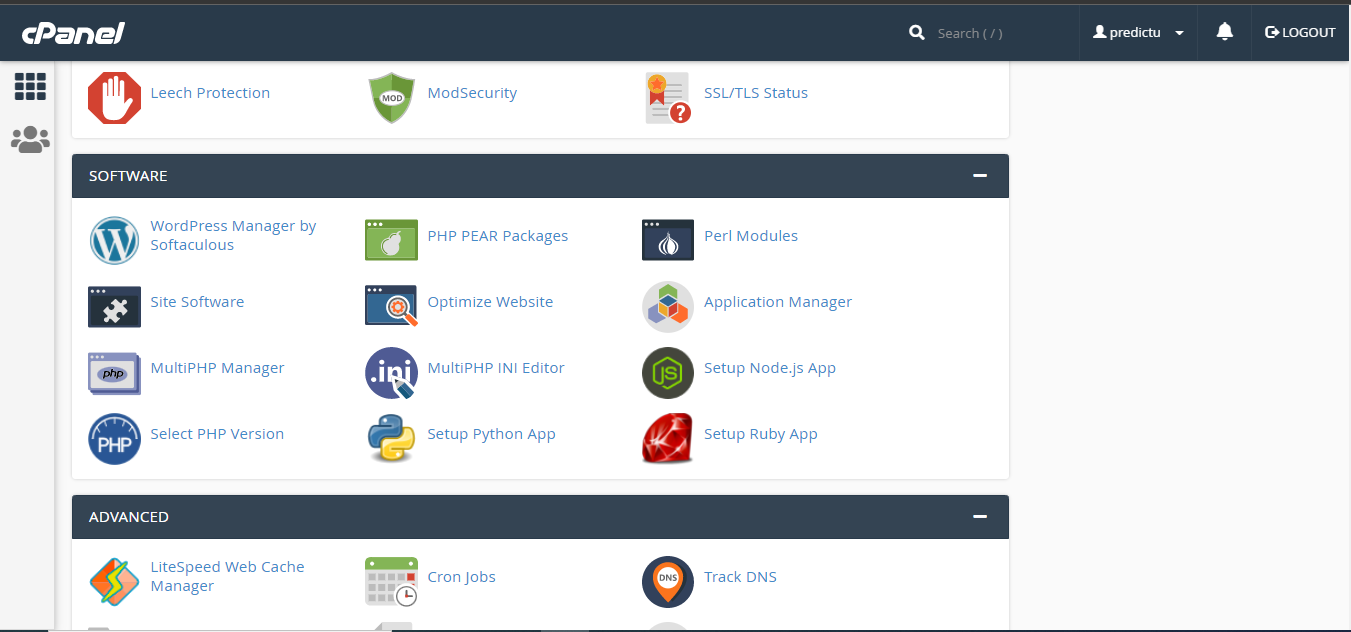


Fig 12 : c-pannel 2

**4.12 Backend**

* [Scripting languages](https://en.wikipedia.org/wiki/Scripting_language) like [PHP](https://en.wikipedia.org/wiki/PHP), [Python](https://en.wikipedia.org/wiki/Python_(programming_language)), [Ruby](https://en.wikipedia.org/wiki/Ruby_(programming_language)), [Perl](https://en.wikipedia.org/wiki/Perl), [Node.js](https://en.wikipedia.org/wiki/Node.js), or [Compiled languages](https://en.wikipedia.org/wiki/Compiled_language) like [C#](https://en.wikipedia.org/wiki/C_Sharp_(programming_language)), [Java](https://en.wikipedia.org/wiki/Java_(programming_language)) or [Go](https://en.wikipedia.org/wiki/Go_(programming_language))
* [Automated testing frameworks](https://en.wikipedia.org/wiki/Automated_Testing_Framework) for the language being used
* [Application Data Access](https://en.wikipedia.org/w/index.php?title=Application_Data_Access&action=edit&redlink=1)
* [Application Business Logic](https://en.wikipedia.org/w/index.php?title=Application_Business_Logic&action=edit&redlink=1)
* [Database administration](https://en.wikipedia.org/wiki/Database_administrator)
* [Scalability](https://en.wikipedia.org/wiki/Scalability)
* [High availability](https://en.wikipedia.org/wiki/High_availability)
* Security concerns, [authentication](https://en.wikipedia.org/wiki/Authentication) and [authorization](https://en.wikipedia.org/wiki/Authorization)
* [Software Architecture](https://en.wikipedia.org/wiki/Software_Architecture)
* [Data transformation](https://en.wikipedia.org/wiki/Data_transformation)
* [Backup](https://en.wikipedia.org/wiki/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](https://en.wikipedia.org/wiki/API). In the case of [web](https://en.wikipedia.org/wiki/Web_API) and mobile frontends, the API is often based on [HTTP](https://en.wikipedia.org/wiki/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**

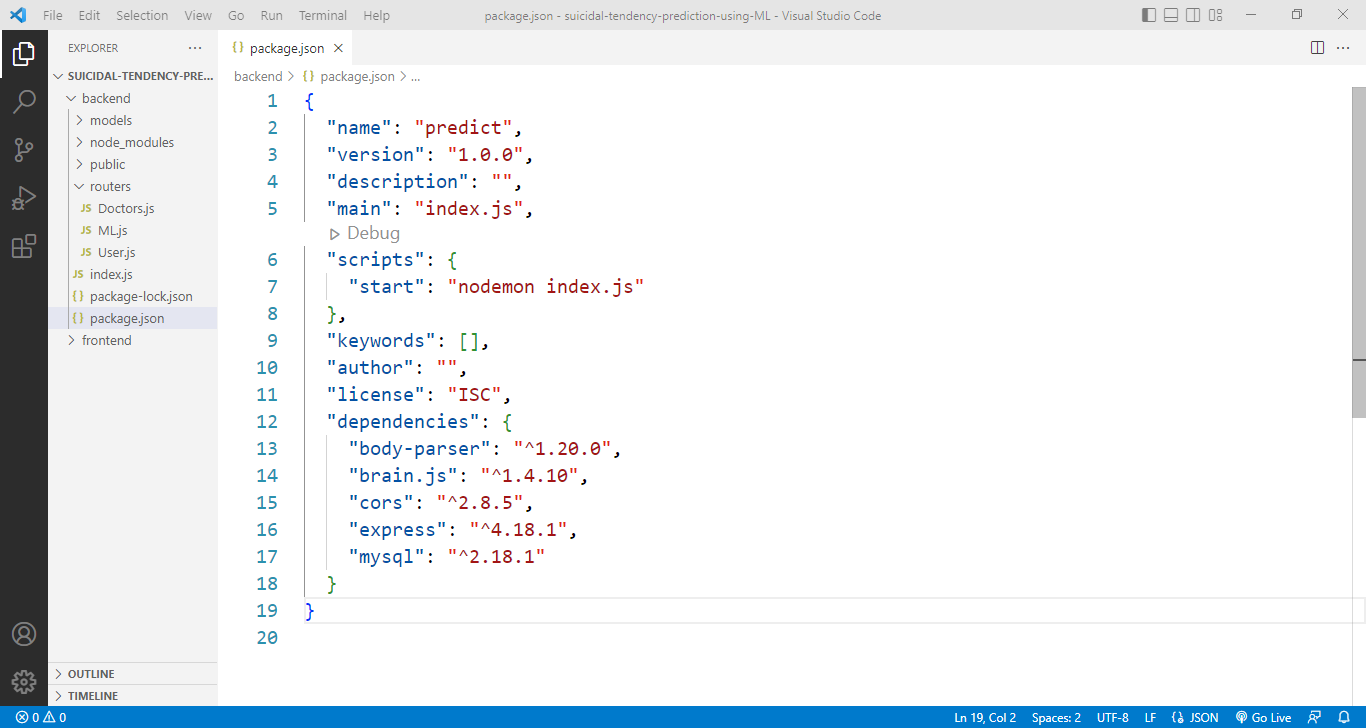
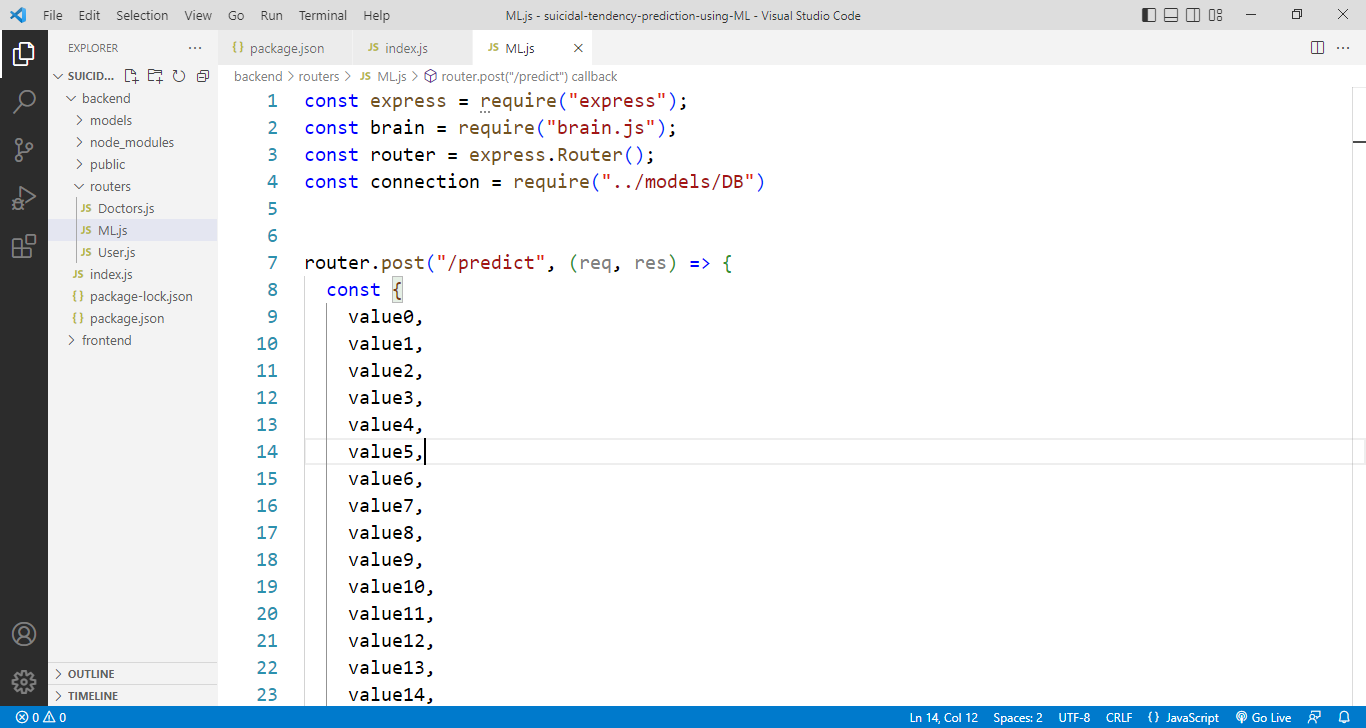
It is Node.js json file. This file includes metadata such as name, author and version as well as starter scripts and dependencies used in the project. Setup node.js server.

Fig 13 : Backend Code 1

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

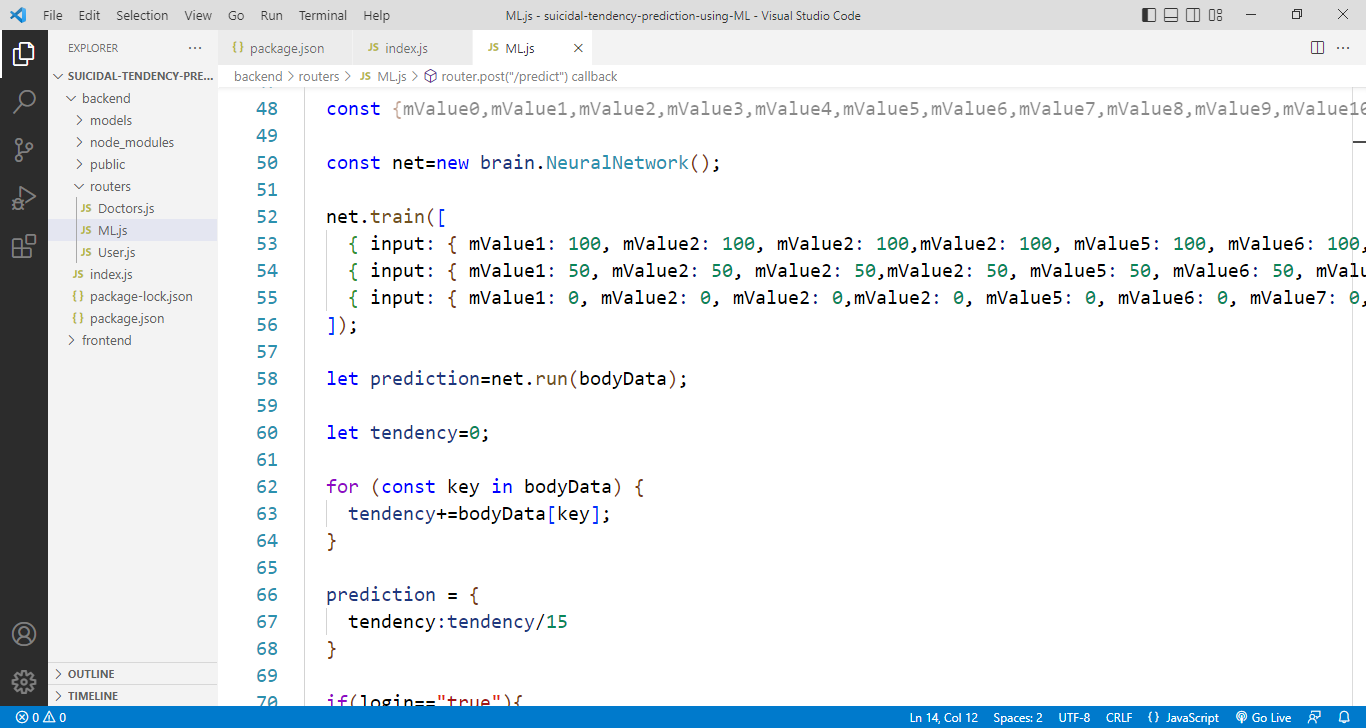


Fig 14 : Backend Code 2

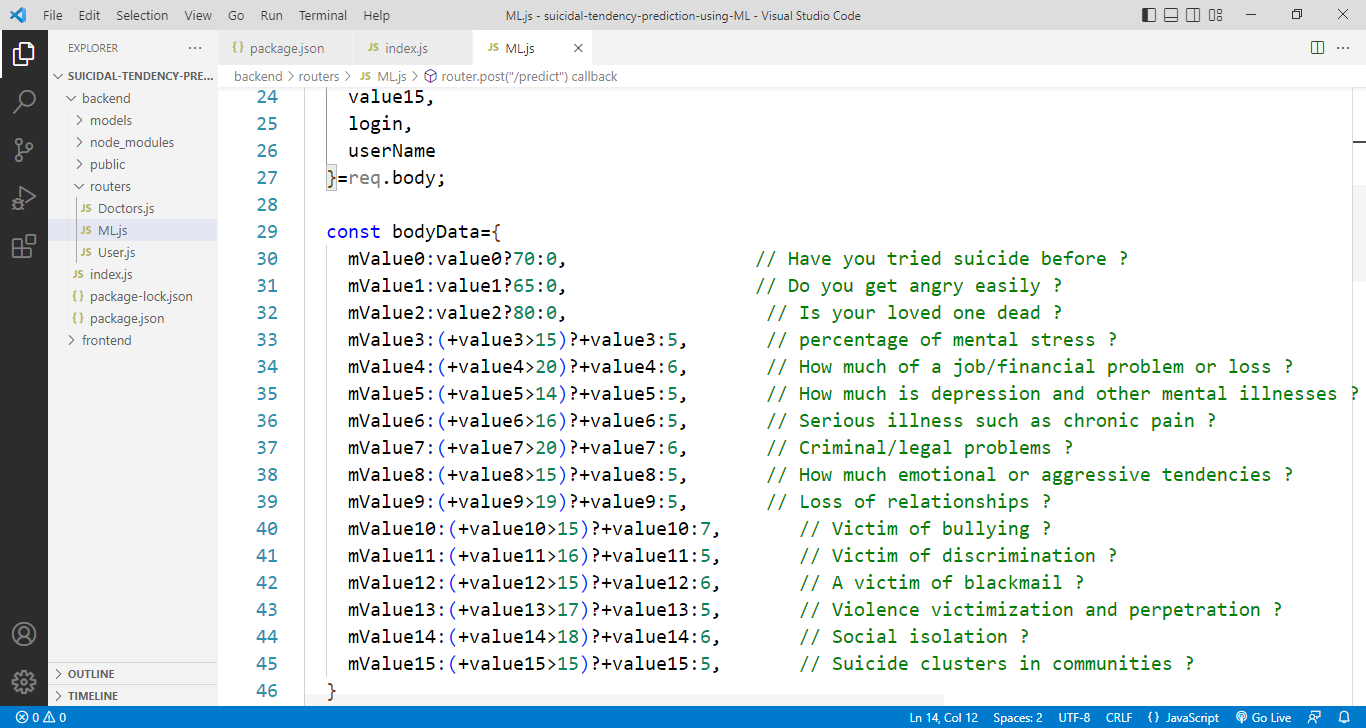
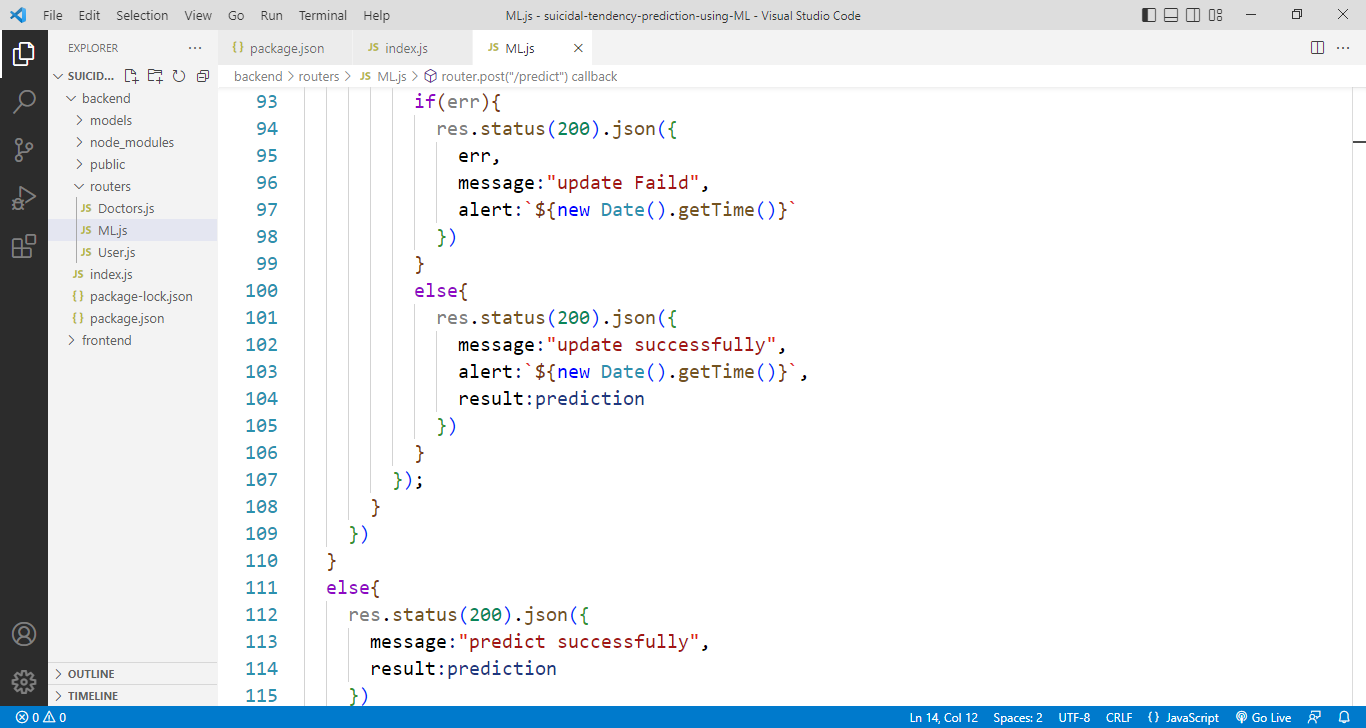
Creating a REST API backend using node.js, express.js. This code frontend input data receive and manae.

Fig 15 : Backend Code 3

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

Fig 16 : Backend Code 4

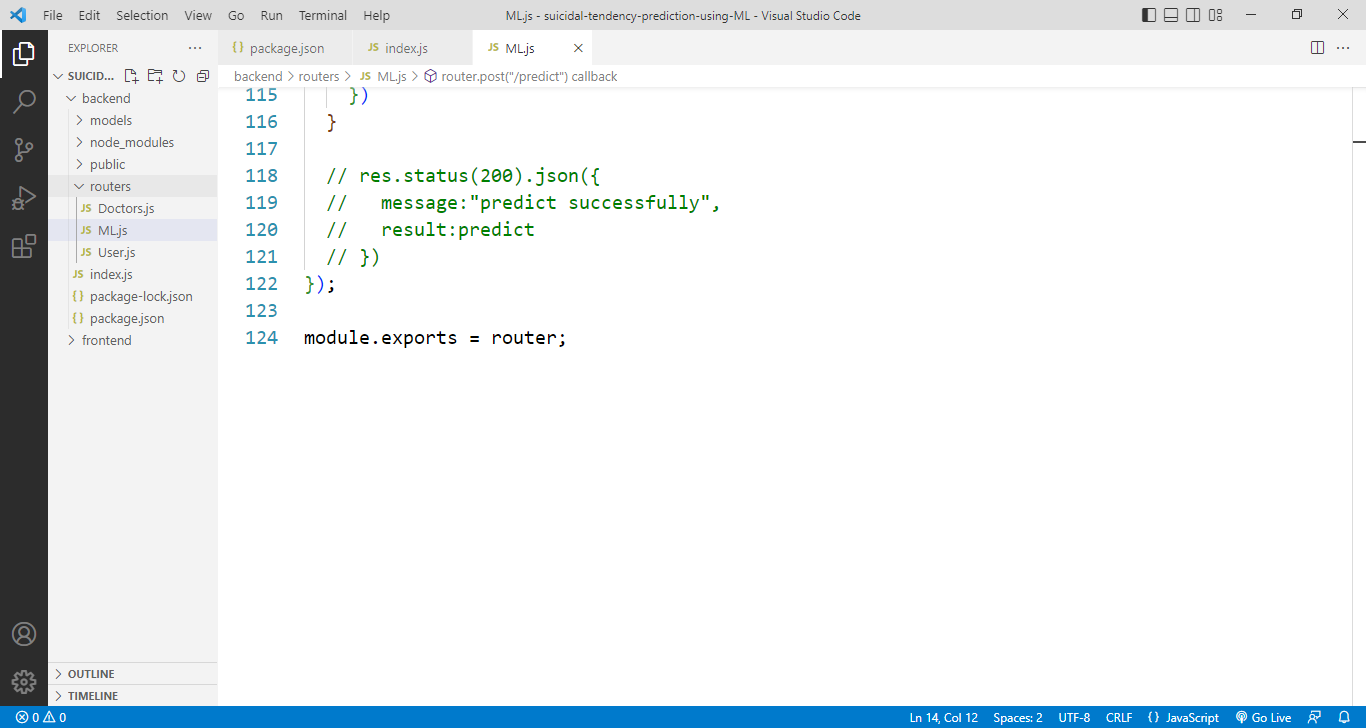
The module.exports is a special object which is included in every JavaScript file in the Node.js application by default.

Fig 17 : Backend Code 5

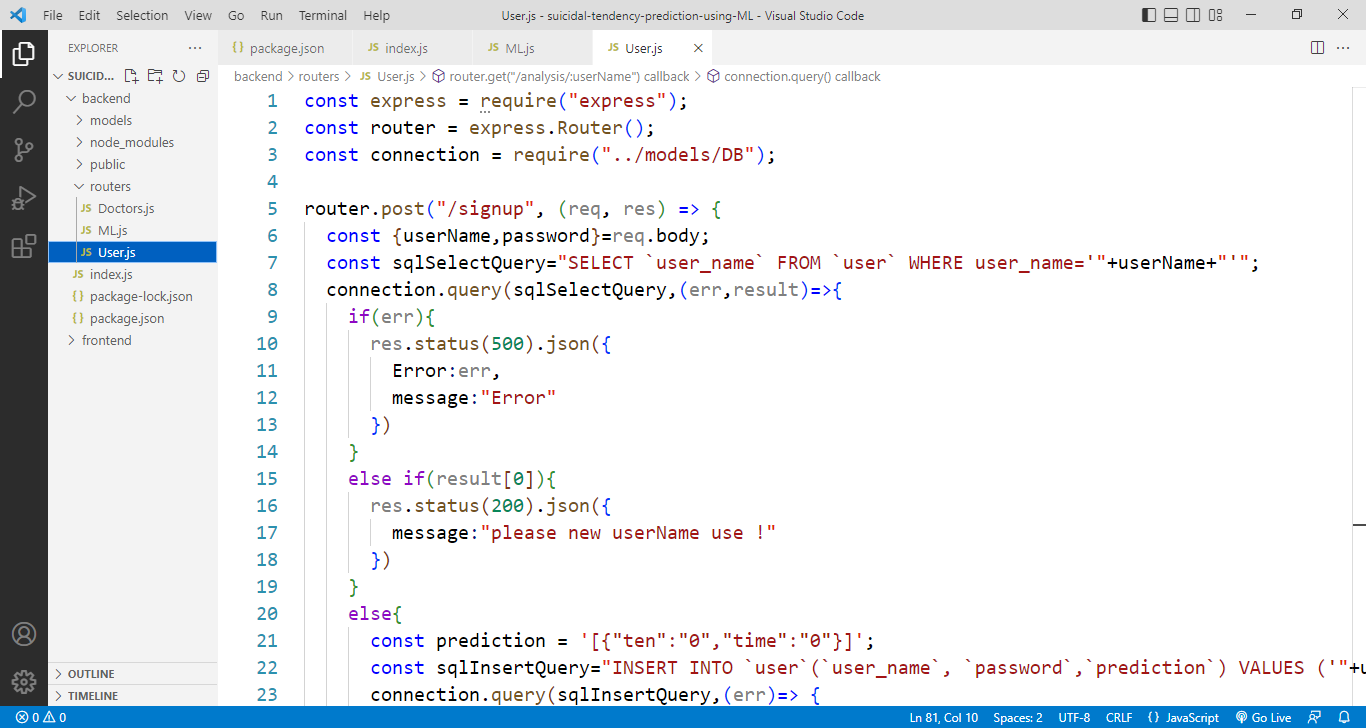
This is backend user router code.

Fig 18 : Backend Code 6

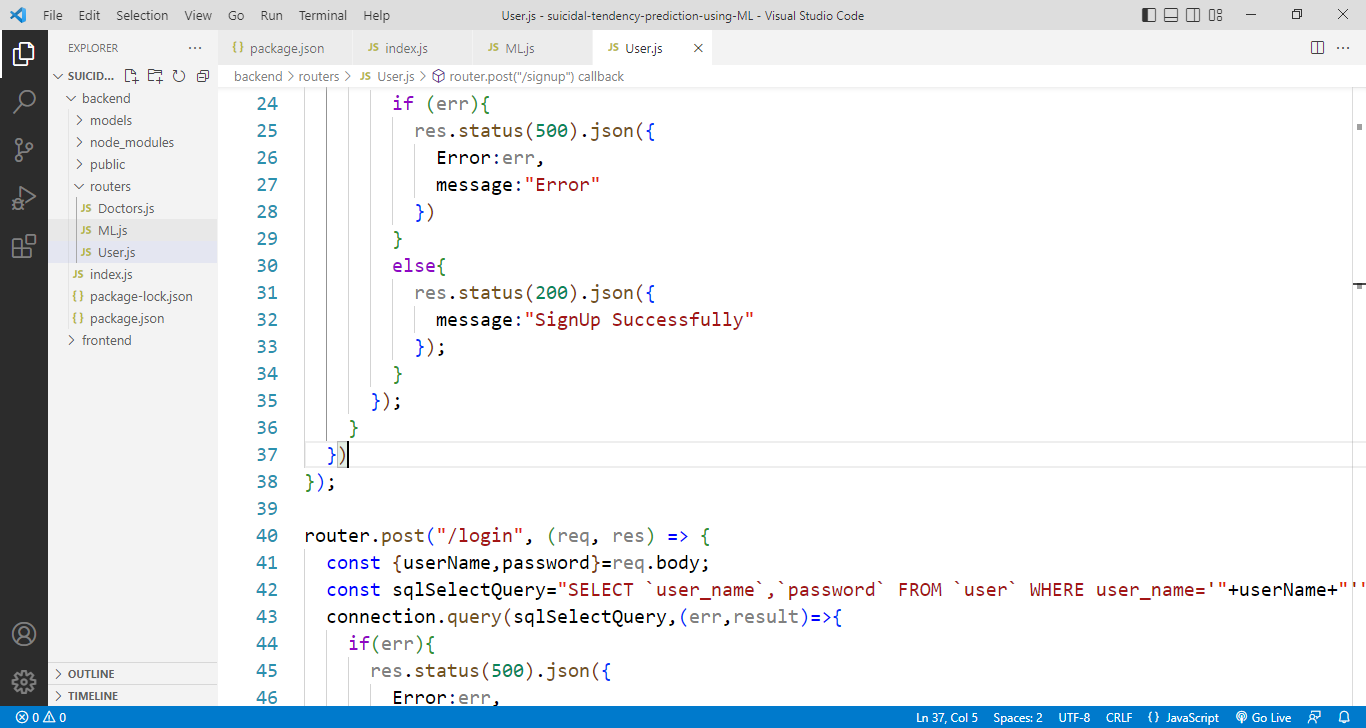
This is backend user login router code. In this code frontend request accept and valid user check. If user name and password match then login

Fig 19 : Backend Code 7

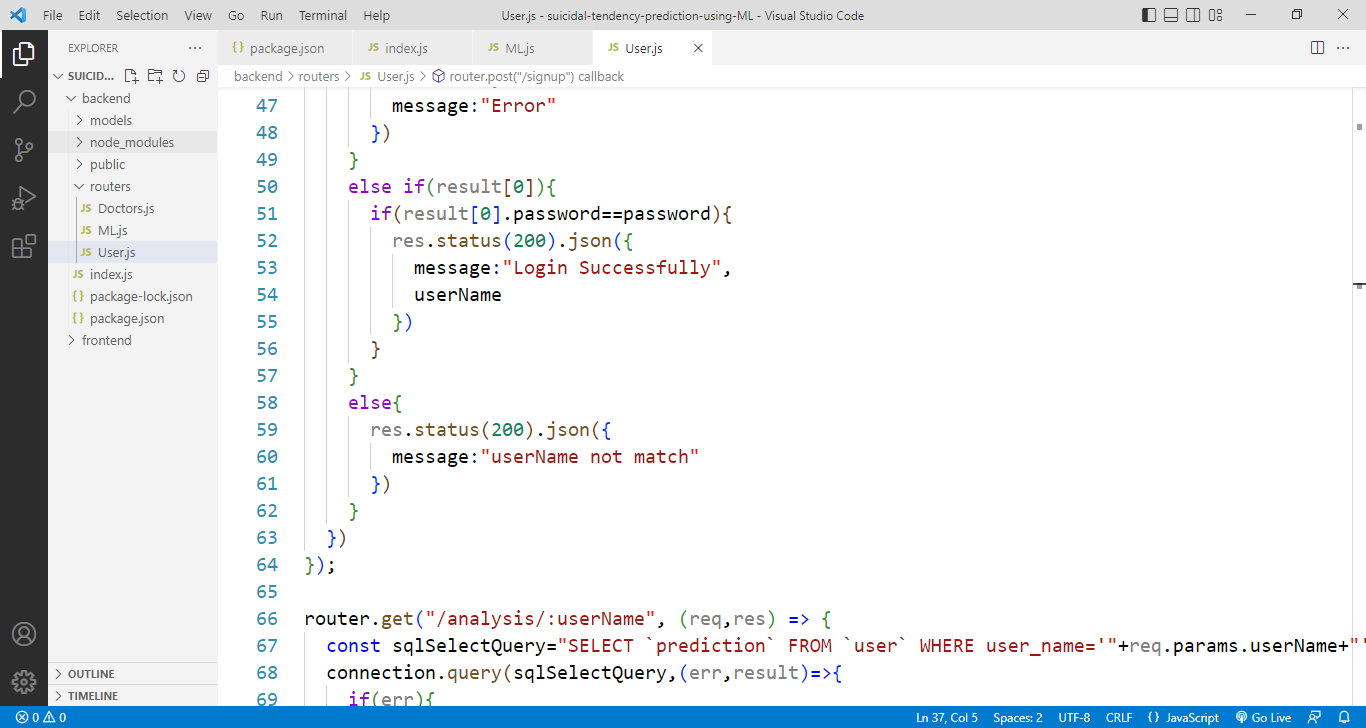
This code check password.

Fig 20 : Backend Code 8

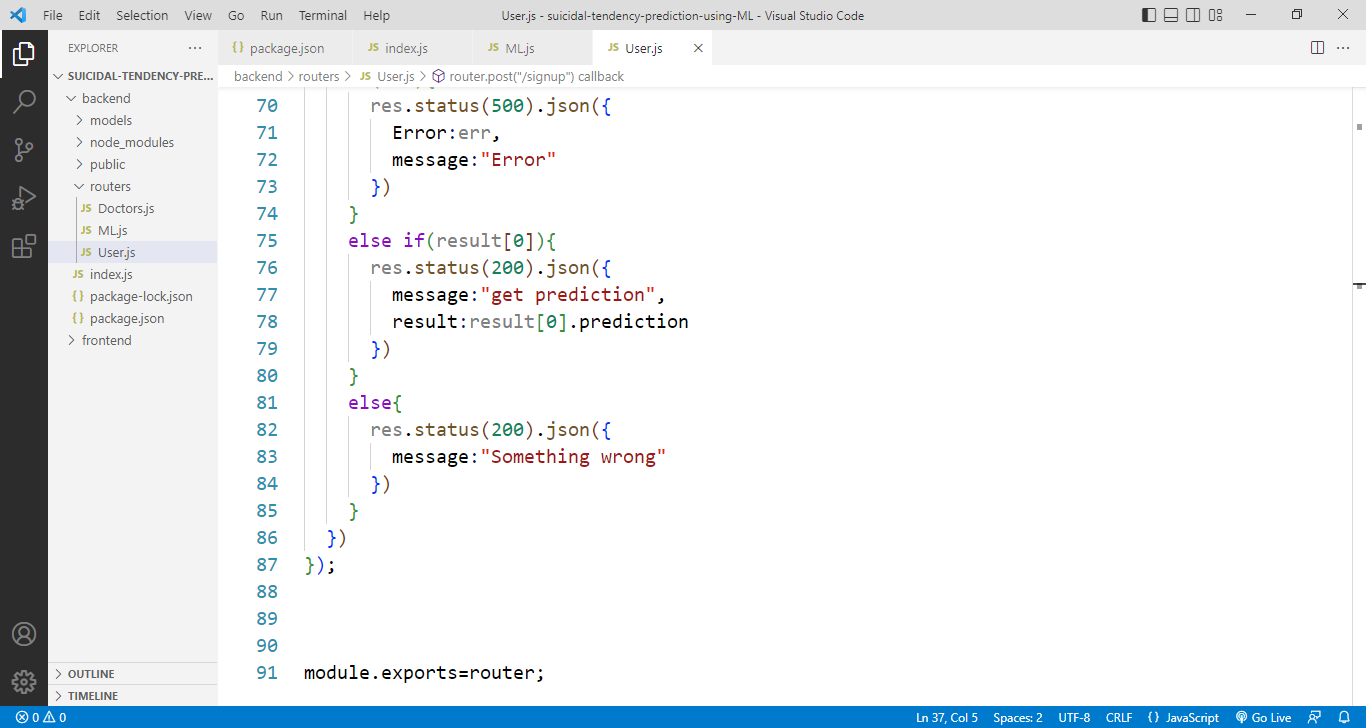
This code is backend analysis router. In this code frontend request accept and analysis.

Fig 21 : Backend Code 9

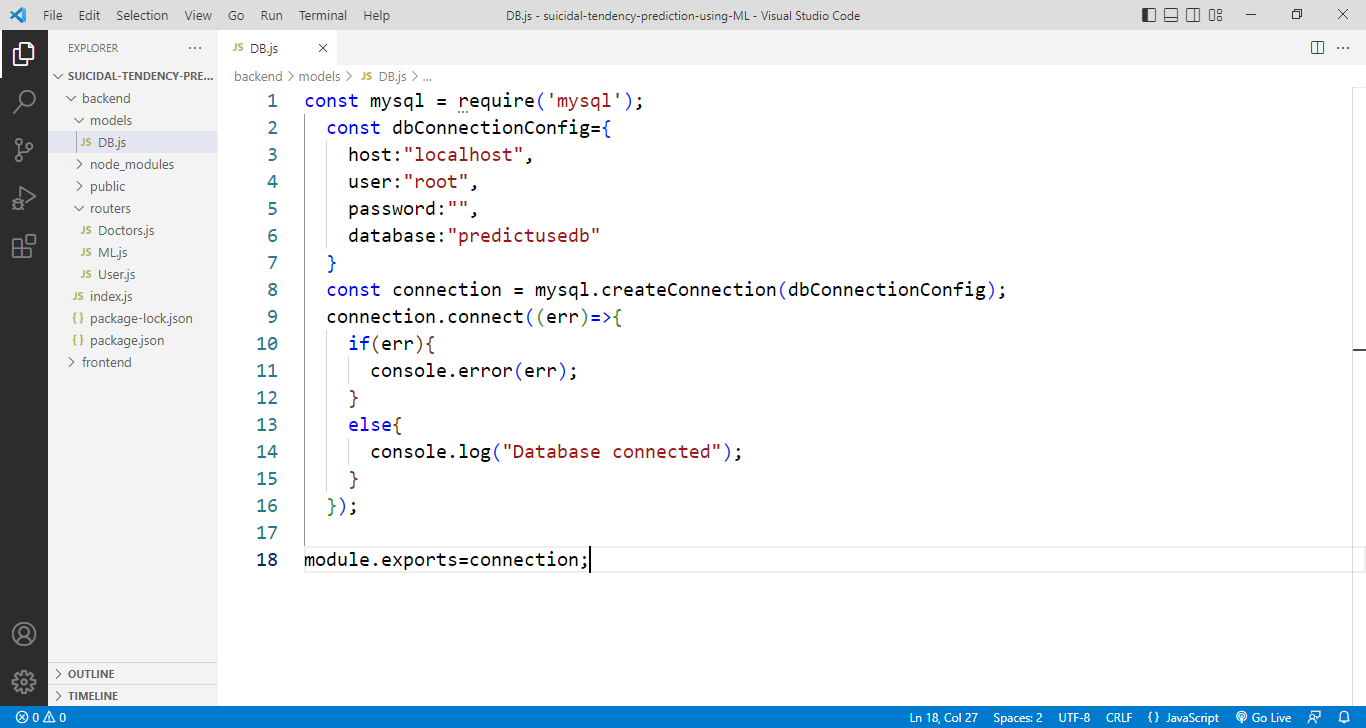
This is backend database connection code.

Fig 22 : Backend Code 10

**4.14 Backend Deploy**

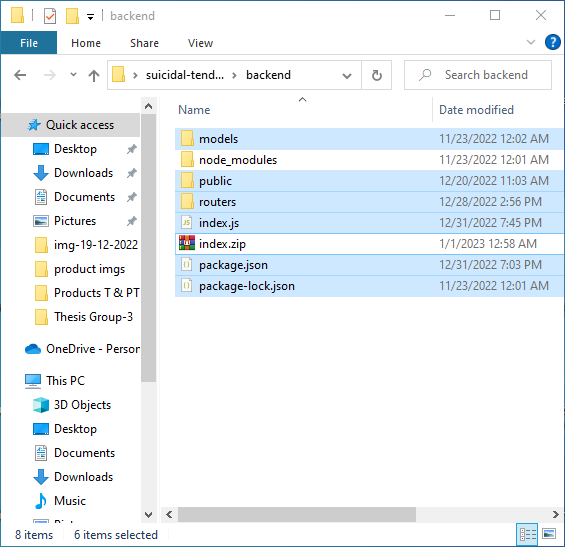
At fast node.js folders and file convert to zip > c-pannel > file manager > public\_html/userbackendapi upload > convert zip to unzip. This backend produce api.

Fig 23 : Backend Files

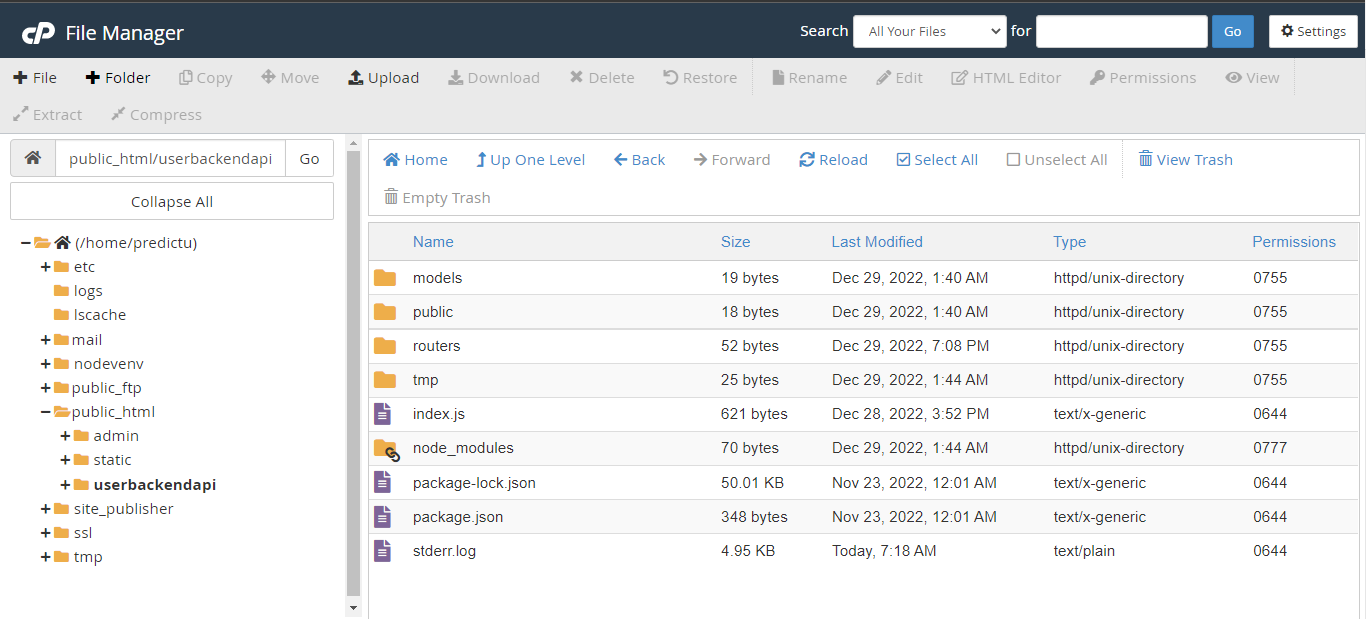


Fig 24 : Backend c-pannel Files

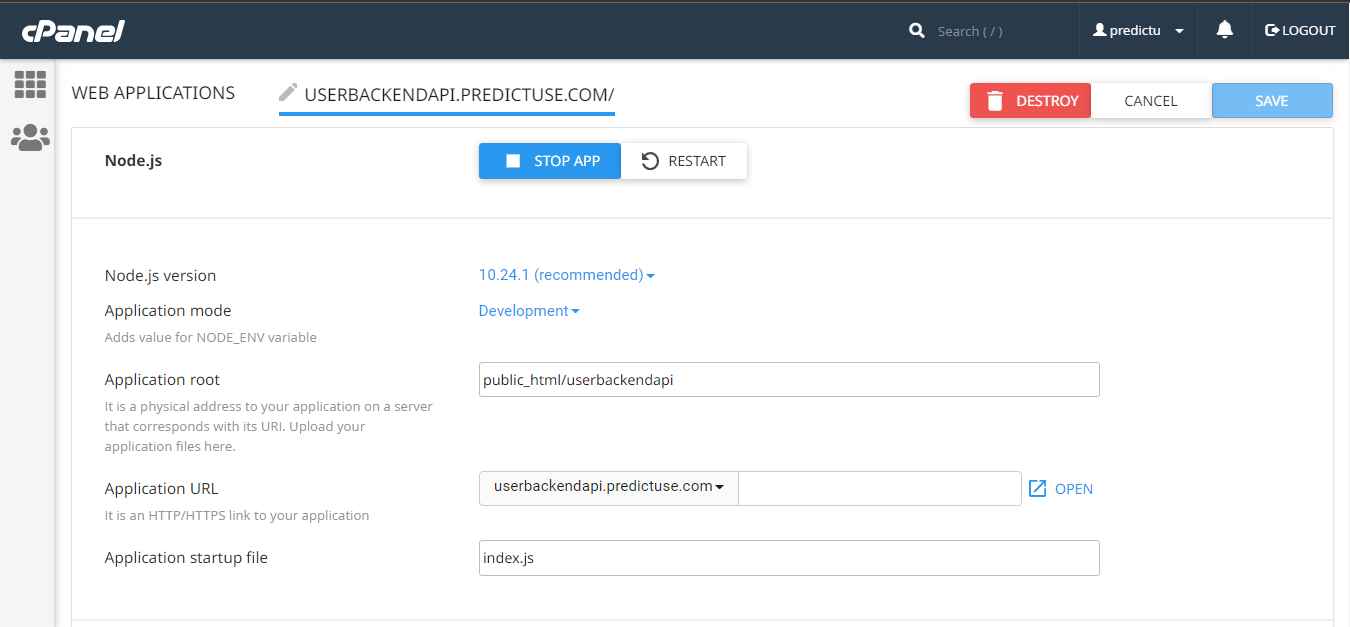
This is node.js code upload process at c-pannel. Node.js folder path set and package file run then start.

Fig 25 : c-pannel nodejs1

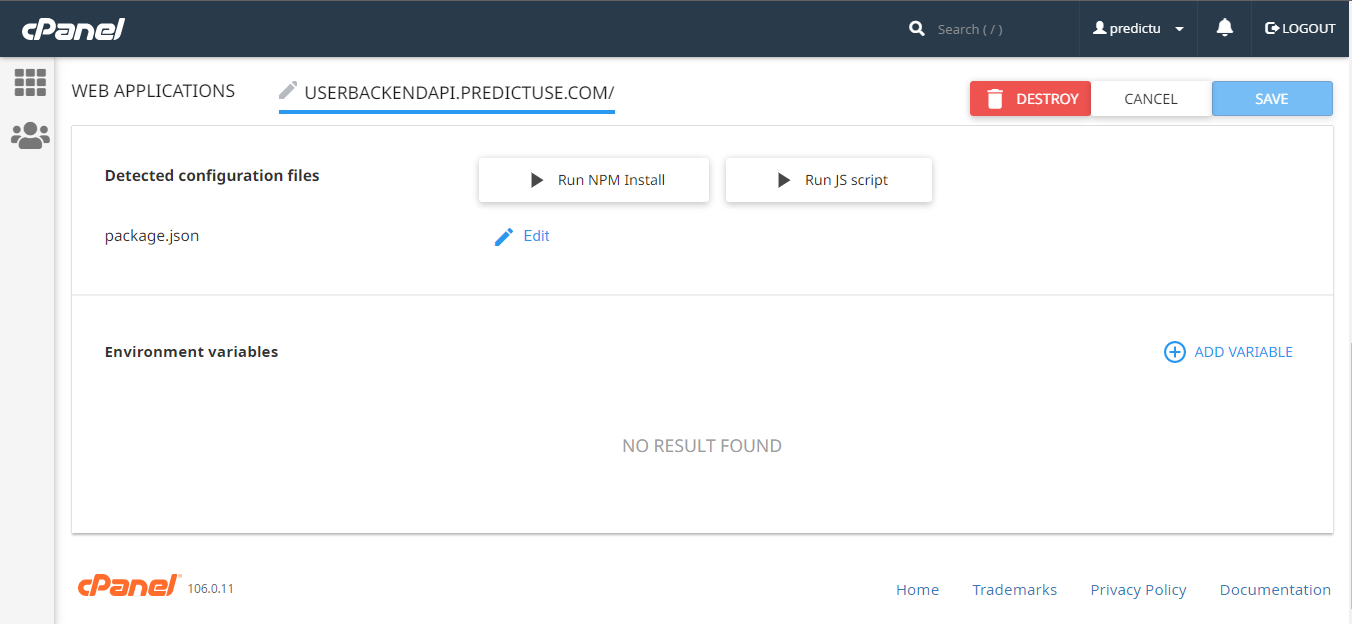
In this all package install of project. And environment variable set.

Fig 26 : c-pannel nodejs2

**4.15 Frontend**

* Markup and web languages such as [HTML](https://en.wikipedia.org/wiki/HTML), [CSS](https://en.wikipedia.org/wiki/CSS), [JavaScript](https://en.wikipedia.org/wiki/JavaScript), and ancillary libraries commonly used in those languages such as [Sass](https://en.wikipedia.org/wiki/Sass_(stylesheet_language)) or [jQuery](https://en.wikipedia.org/wiki/JQuery)
* [Asynchronous](https://en.wikipedia.org/wiki/Asynchronous_I/O) request handling and [AJAX](https://en.wikipedia.org/wiki/Ajax_(programming))
* [Single-page applications](https://en.wikipedia.org/wiki/Single-page_application) (with frameworks like [React](https://en.wikipedia.org/wiki/React_(JavaScript_library)), [Angular](https://en.wikipedia.org/wiki/Angular_(web_framework)) or [Vue.js](https://en.wikipedia.org/wiki/Vue.js))
* [Web performance](https://en.wikipedia.org/wiki/Web_performance) (largest contentful paint, time to interactive, 60 [FPS](https://en.wikipedia.org/wiki/Frame_rate) animations and interactions, memory usage, etc.)
* [Responsive web design](https://en.wikipedia.org/wiki/Responsive_web_design)
* [Cross-browser](https://en.wikipedia.org/wiki/Cross-browser) compatibility issues and workarounds
* [End-to-end testing](https://en.wikipedia.org/wiki/Software_testing) with a [headless browser](https://en.wikipedia.org/wiki/Headless_browser)
* [Build automation](https://en.wikipedia.org/wiki/Build_automation) to transform and bundle JavaScript files, reduce image sizes and other processes using tools such as [Webpack](https://en.wikipedia.org/wiki/Webpack" \o "Webpack) and [Gulp.js](https://en.wikipedia.org/wiki/Gulp.js)
* [Search engine optimization](https://en.wikipedia.org/wiki/Search_engine_optimization)
* [Accessibility](https://en.wikipedia.org/wiki/Web_accessibility) concerns
* Basic usage of image editing tools such as [GIMP](https://en.wikipedia.org/wiki/GIMP) or [Photoshop](https://en.wikipedia.org/wiki/Adobe_Photoshop)
* [User Interface](https://en.wikipedia.org/wiki/User_Interface)

**4.16 Frontend Code**

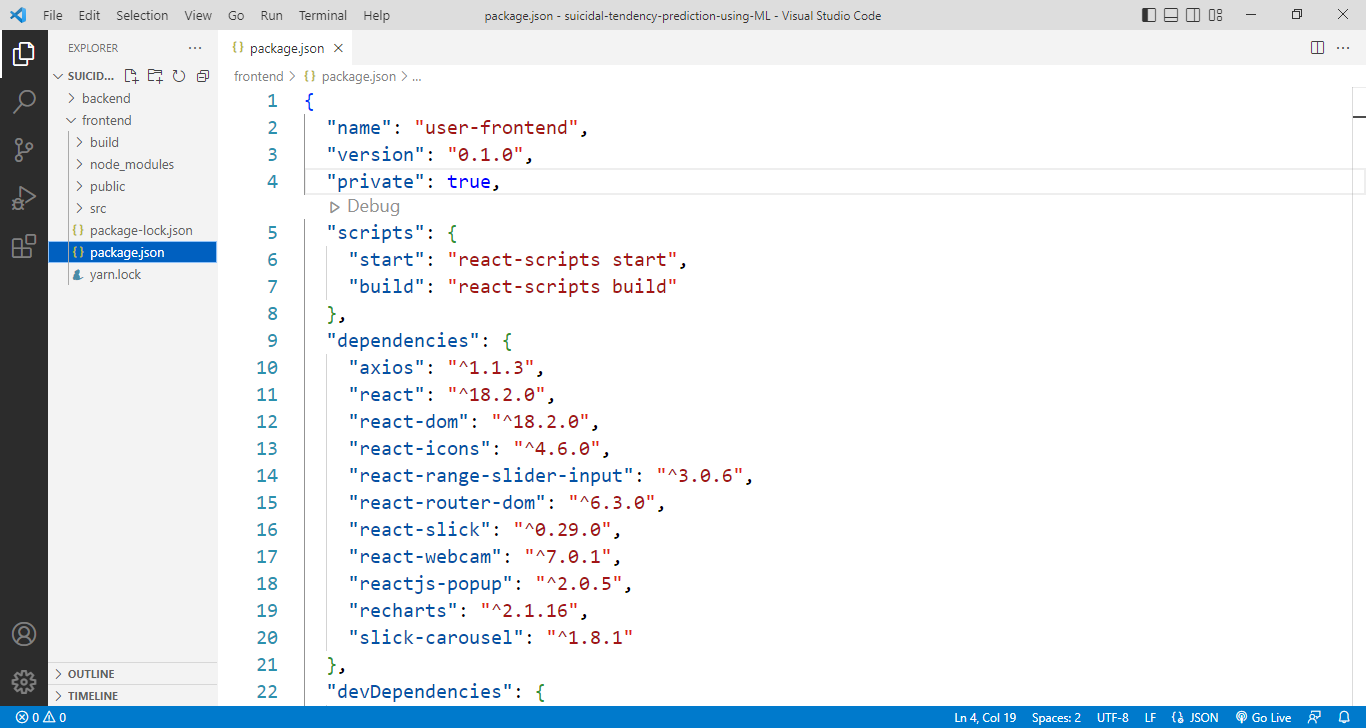
It is React.js json file. This file includes metadata such as name, author and version as well as starter scripts and dependencies used in the project.

Fig 27 : Frontend code 1

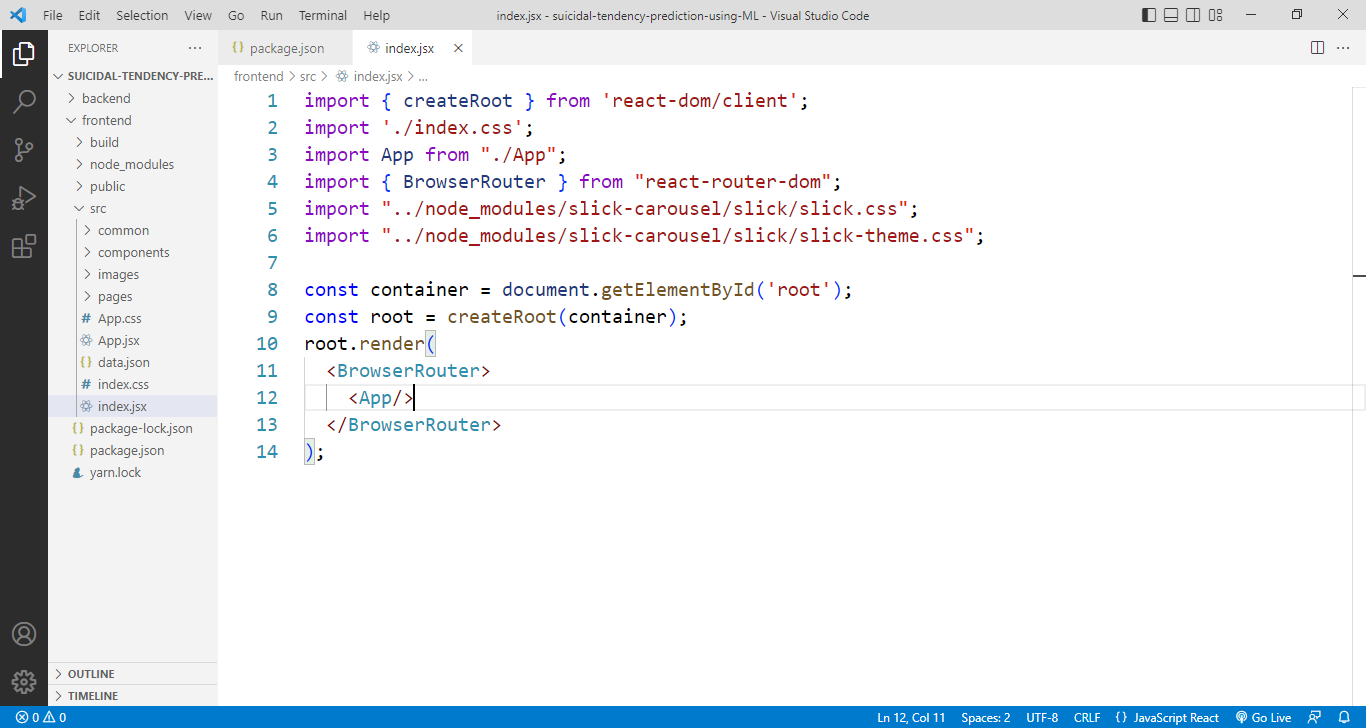
This is React.js index.js code.

Fig 28 : Frontend code 2

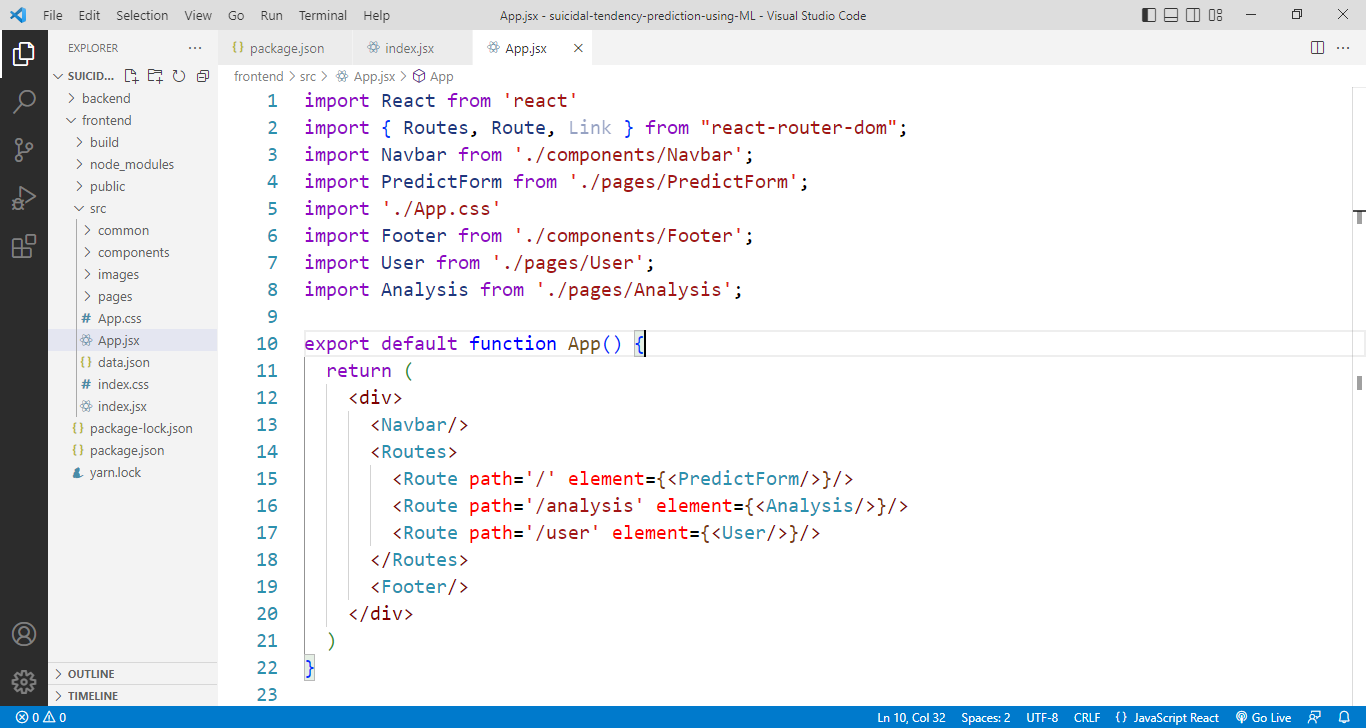
This is App.js file code. In this code all react js file manage and is section manage routing.

Fig 29 : Frontend code 3

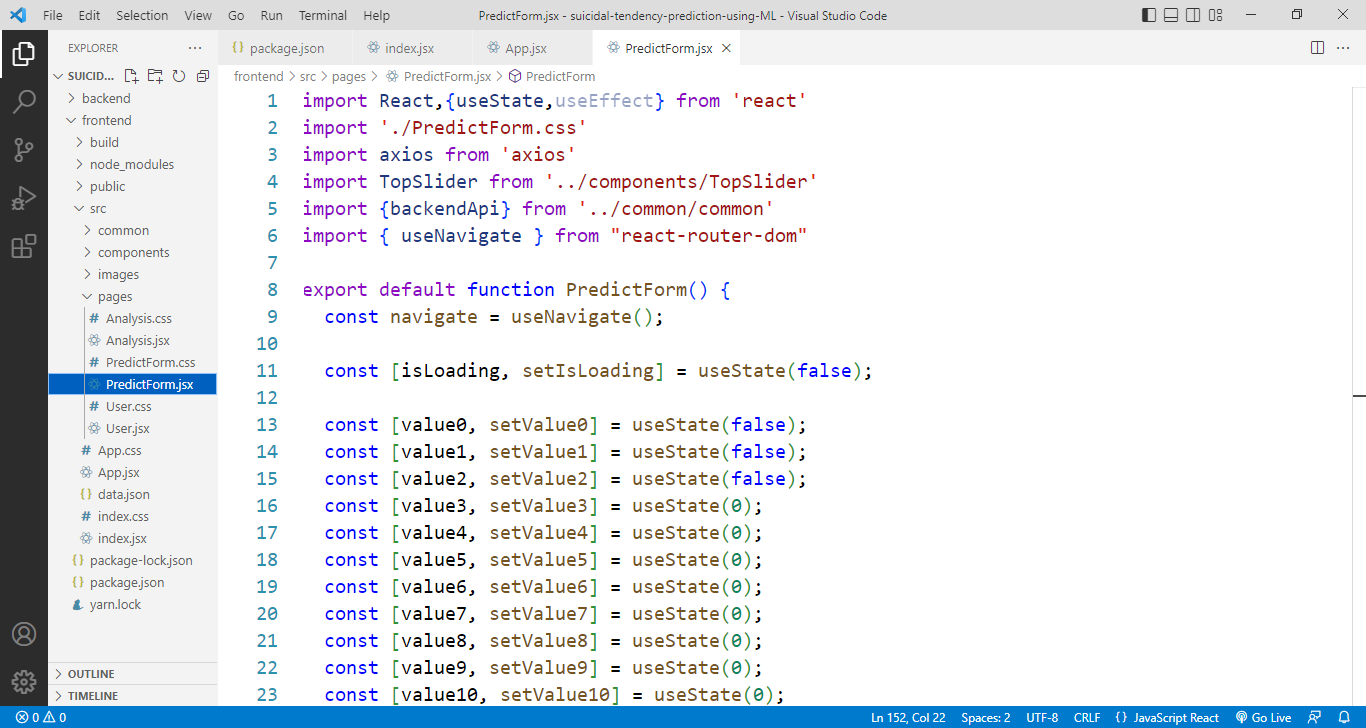
This react file is predict input data ui design and request backend with predict data.

Fig 30 : Frontend code 4

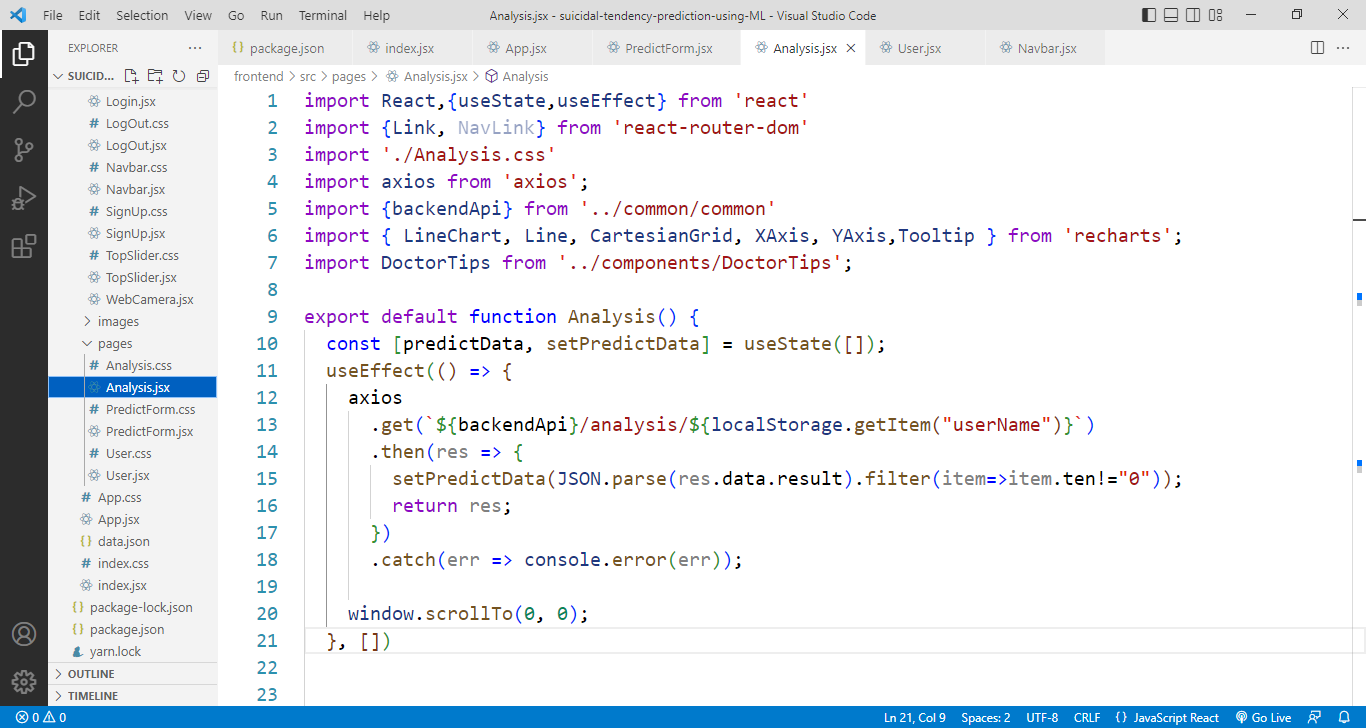
In this react file analysis part and analysis ui desing. This code accept backend data and show chart.

Fig 31 : Frontend code 5

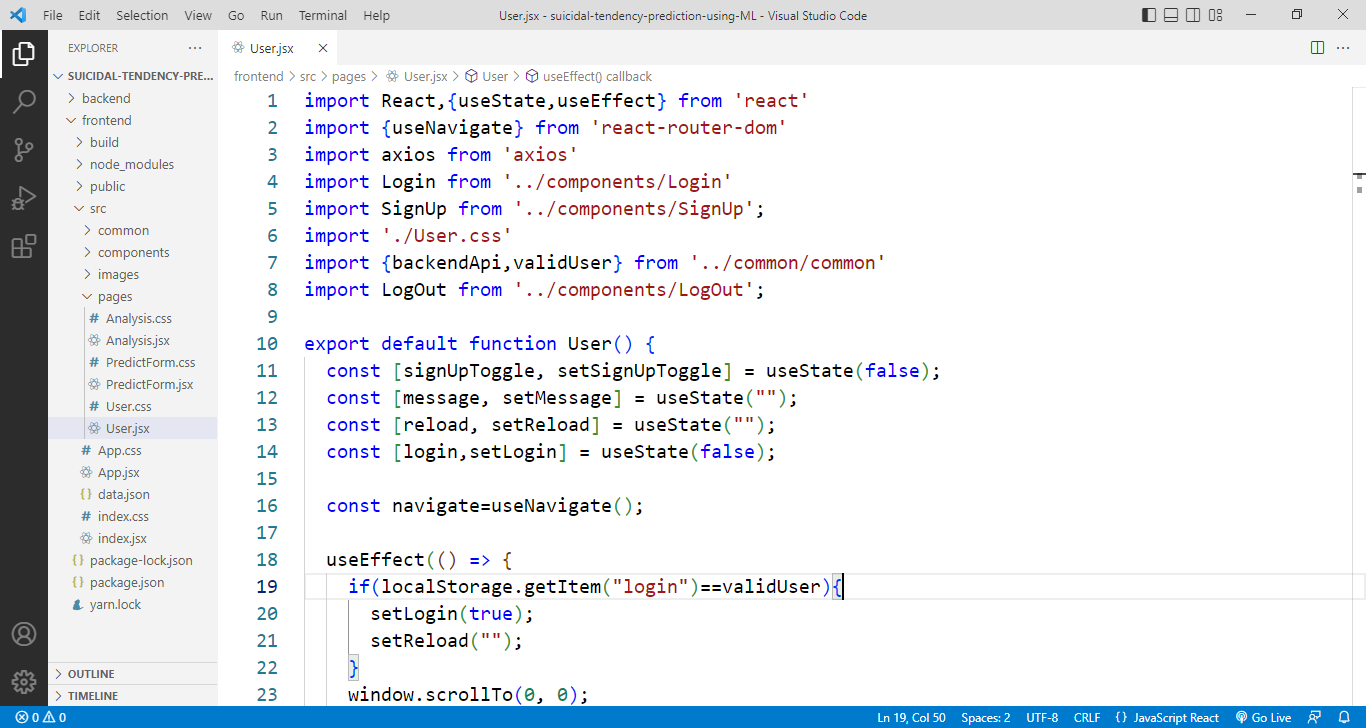
This route user login and signup ui. User data communicate backend.

Fig 32 : Frontend code 6

**4.17 Frontend Deploy**

At fast React Code, Build > zip > c-pannel > public\_html folder > upload > unzip

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

$ npm run build

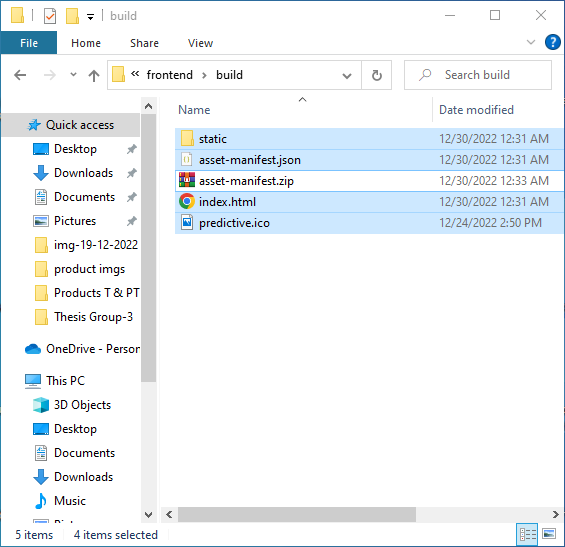


Fig 33 : Frontend Files

**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");

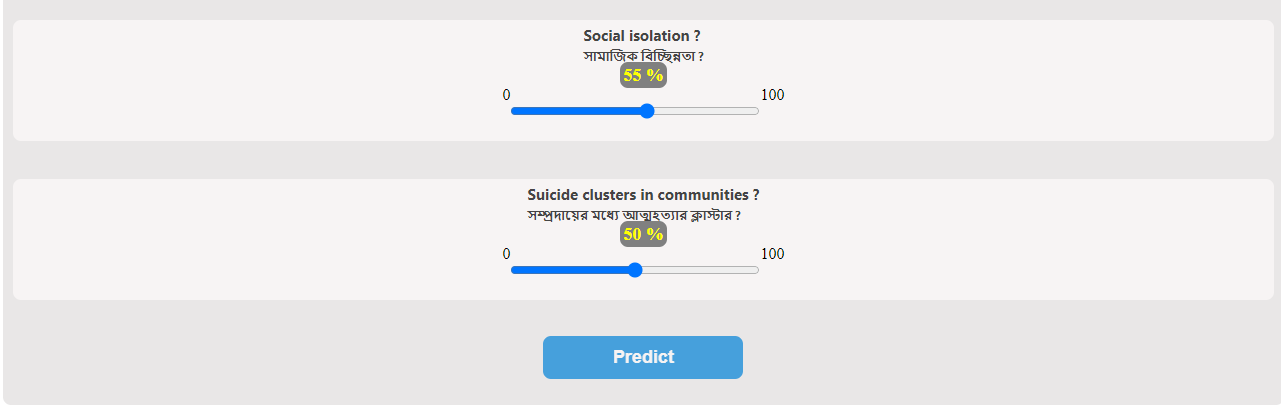
          setIsLoading(false);

          window.scrollTo(0, 0);

          return res;

        })

        .catch(err => console.error(err));

****    }

  }

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);

  }, [])

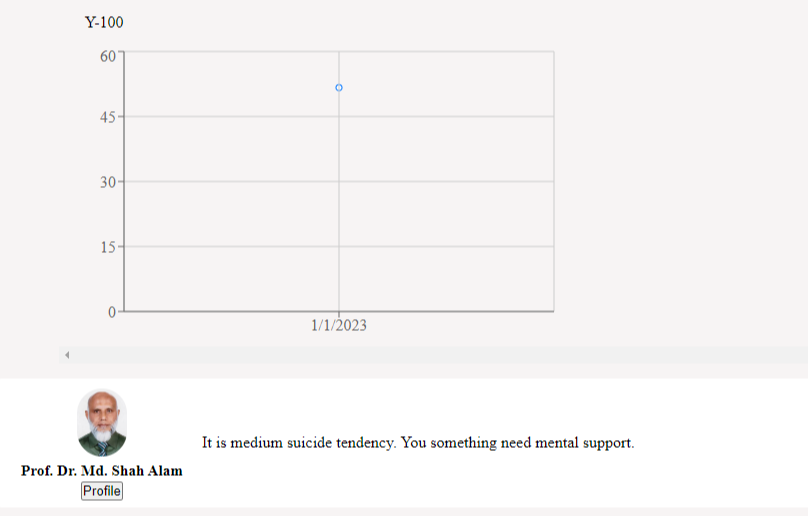


Fig 35 : Predict Result 1

**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");

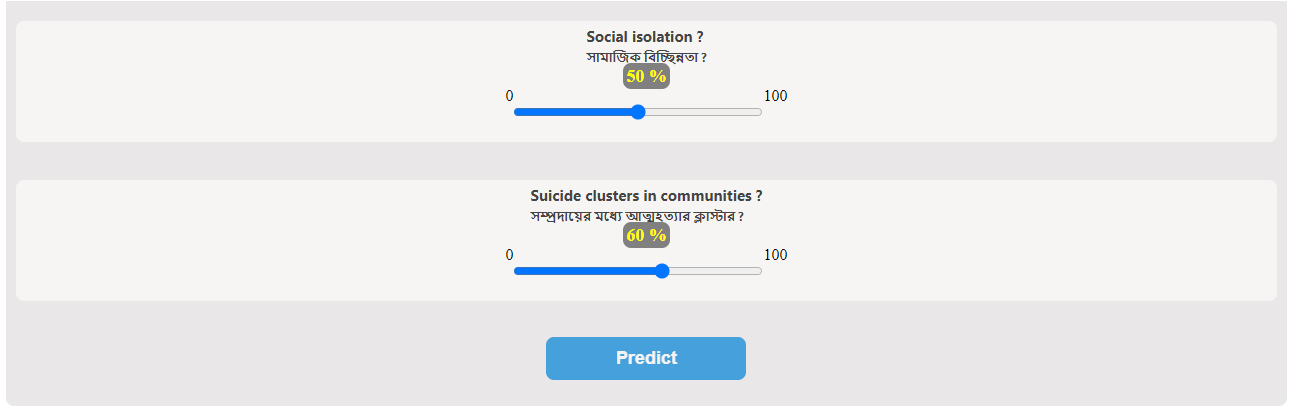
          setIsLoading(false);

          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);

  }, [])

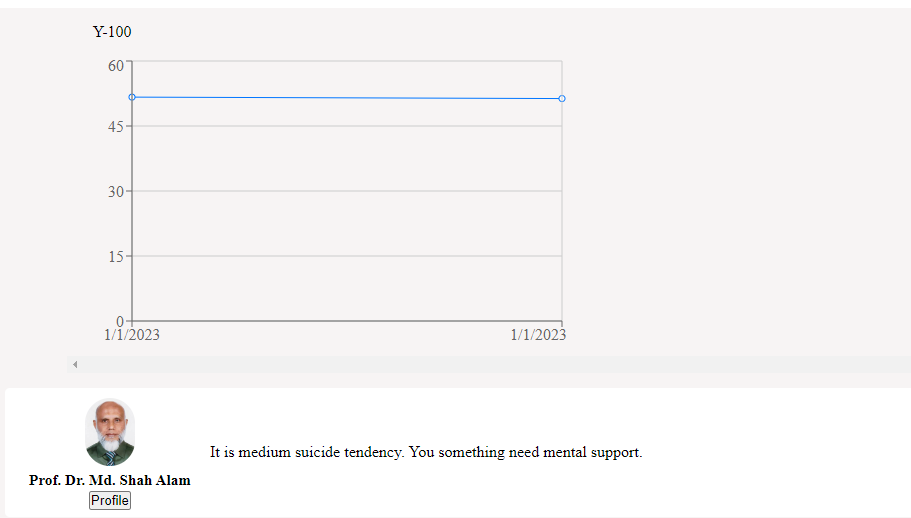
****

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");

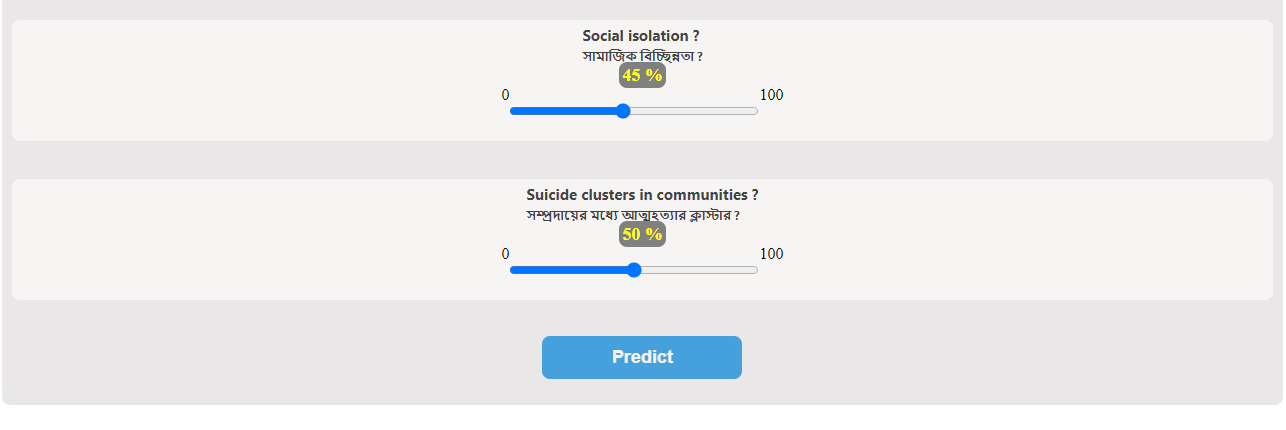
          setIsLoading(false);

          window.scrollTo(0, 0);

          return res;

        })

        .catch(err => console.error(err));

****    }

  }

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);

  }, [])

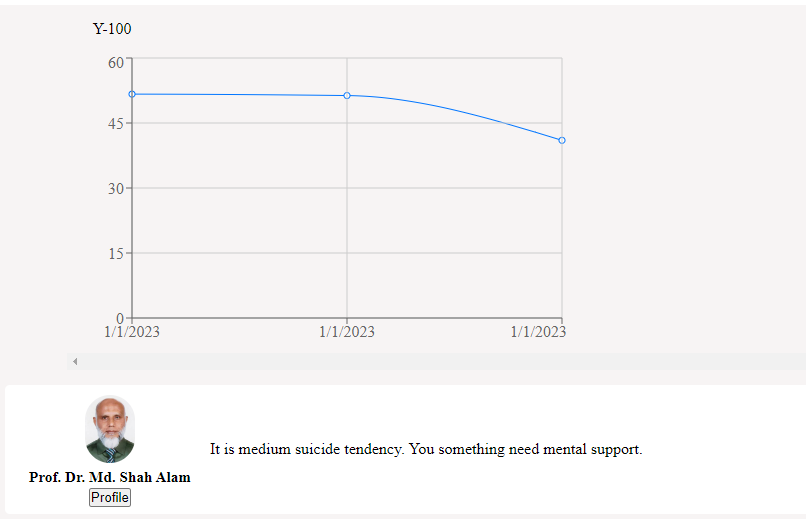
****

Fig 39 : Predict Result 3

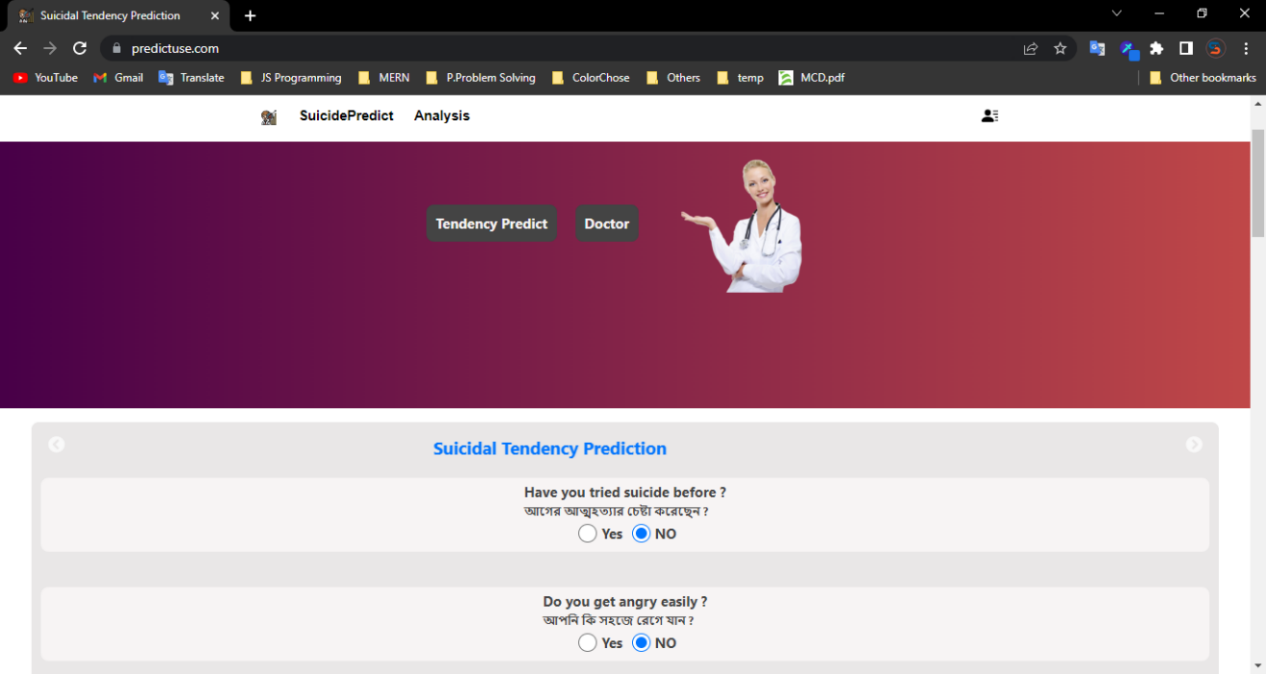
**4.19 Finally Complete Our Siucidal Tendency Prediction Web Site**

Fig 40 : WebSite Home Page

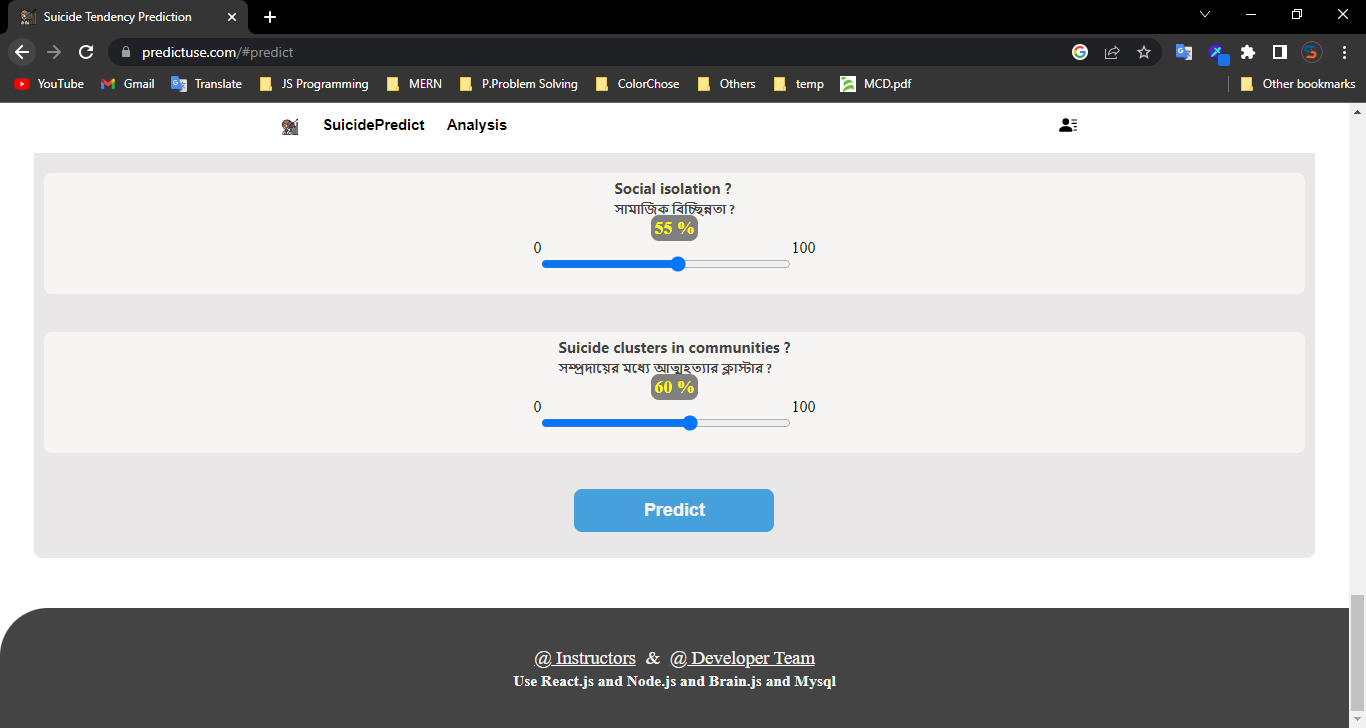
In this question data input and predict.

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.

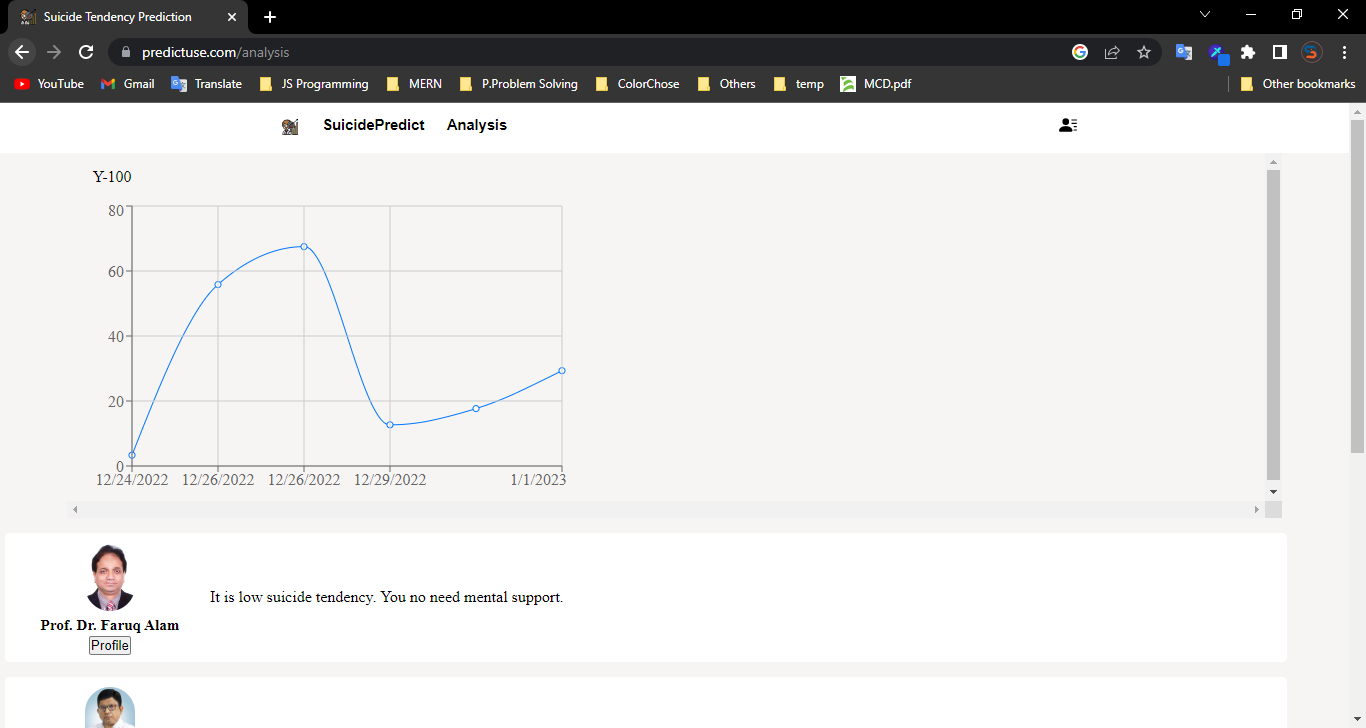


Fig 42 : WebSite Result Page

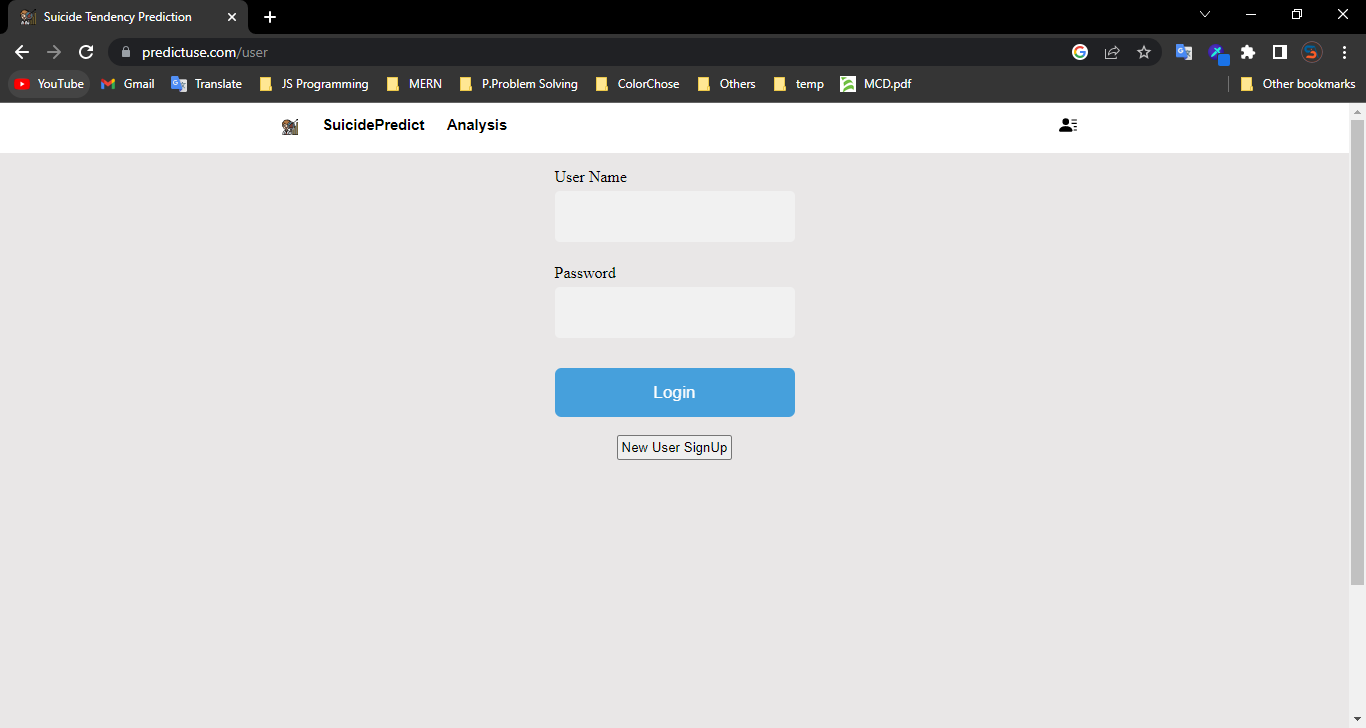
In this login and signUp page. At first signUp for need user name and password.then login.

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.

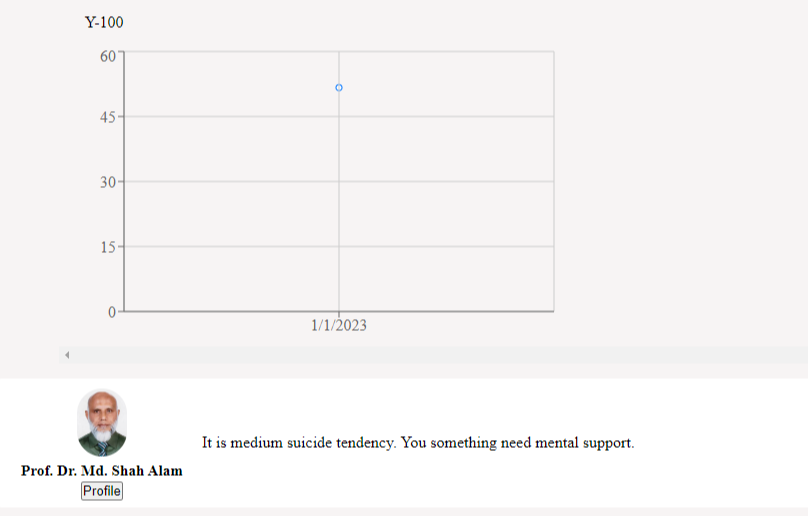
Suicide tendency is 51.66%.

Fig 44 : Result 1

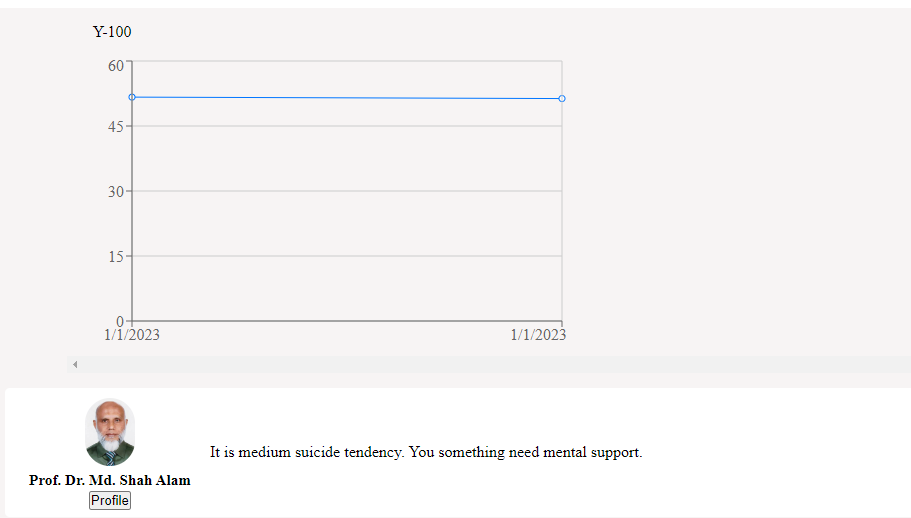
Suicide tendency is 51.33%.

Fig 45 : Result 2

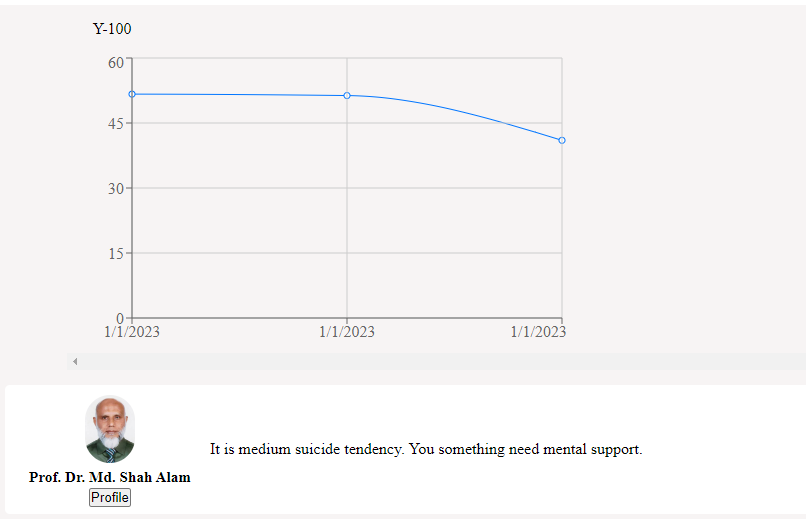
Suicide tendency is 41%.

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](https://psychiatryonline.org/doi/10.1176/appi.ajp-rj.2017.120105?utm_source=TrendMD&utm_medium=cpc&utm_campaign=American_Journal_of_Psychiatry_Residents%27_Journal_TrendMD_0).

[G Rakesh](https://scholar.google.com/citations?user=ddRD3BYAAAAJ&hl=en&oi=sra) - 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>