ProEDA

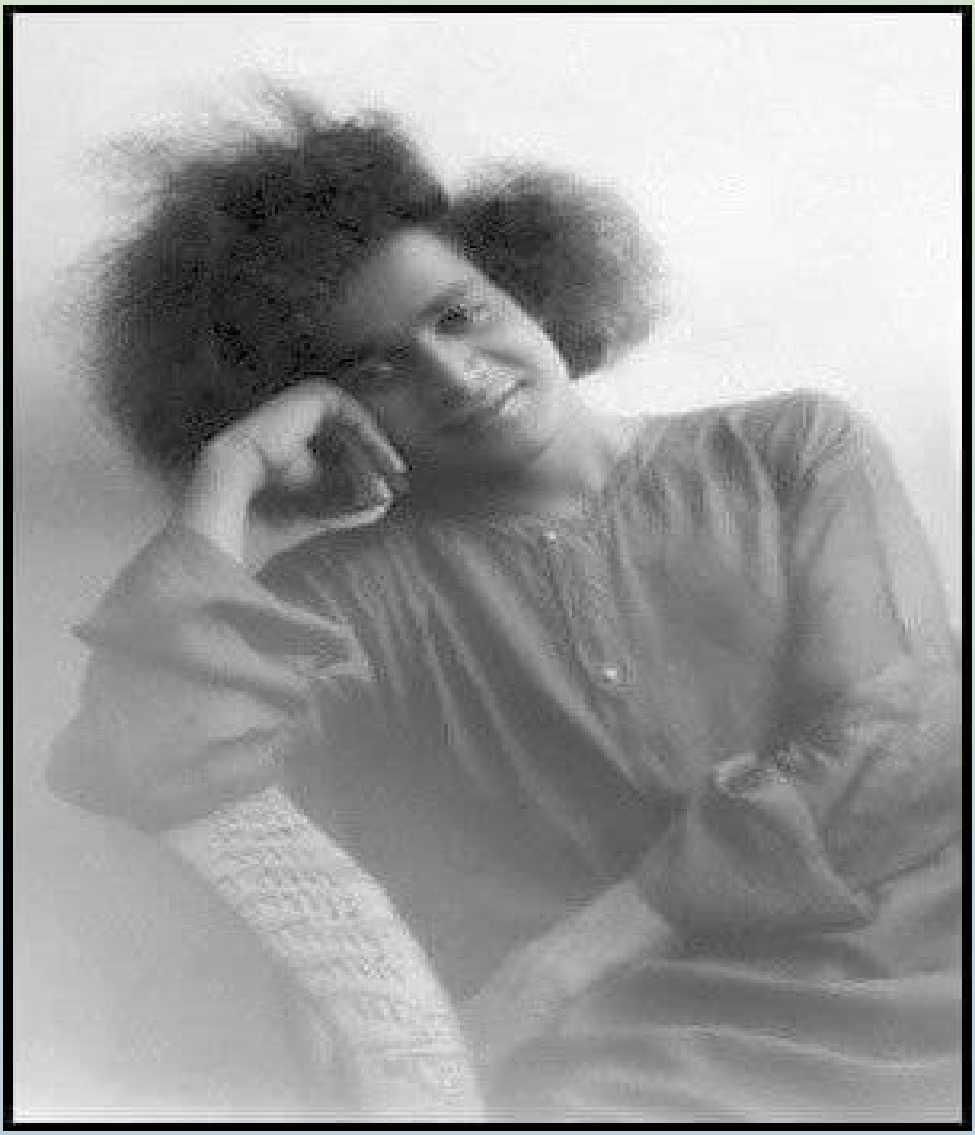
(Project Exploratory Data Analysis)

GROUP PROJECT 2020-21

submission by :
PARTH
184403
III BCA

Group Members :
PARTH
B.SAIKRISHNA
DEEPAM RAI
SAINATH REDDY

GUIDE:
Dr. Sampath Lonka
Phd.



Dedicated in the Thy Lotus feet of Our Beloved
Swami Sri Sathya Sai

ACKNOWLEDGEMENT

We are grateful to the institute for incorporating such helpful and mind expanding project in our curriculum. We express our heartfelt gratitude to our guide Dr. Sampath Lonka for guiding us through this project and also we thank our senior brothers Raghava, Akhil Sai for giving us the advices as to how to go with the project and also their project demonstration which is akin to ours. And we are grateful to all whose name we missed but who directly or indirectly helped us to bring this project to life.

In view of the bigger picture, I would like to extend my thanks to the internet community, the people in forums like the vue js forum, and stack overflow, for their help and support, replying to every single question that was raised due to ignorance in the new technologiesused in this project. We are forever indebted and grateful to the Creator of Universe, our beloved mother Sai.

CERTIFICATE

Sri Sathya Sai Institute of Higher Learning
(Deemed to be University)
Department of Mathematics & Computer ScienceCertificate

This is to certify that this project report entitled ProEDA being submitted by Parth, III year Bchelor of Computer Applications is a record of bonafide project work carried out by them under my supervision and guidance during the academic year 2020-21 in the Department of Mathematics and Computer Science, Sri Sathya Sai Institute OfHigher Learning, Muddenahalli campus.

Place: .....
Date: .....

.....
Dr. Sampath Lonka
Project Guide

CONTENTS

1. ProEDA
2. Technologies Used
    2.1 Vue framework
    2.2 Vuetify Component LIbrary
    2.3 Node.js
    2.4 Express
    2.5 MongoDB (No-SQL database)
    2.6 Google Charts
    2.7 papa-parse(CSV file to JSON)
3. Development Platform
4. Workflow: how user uses?
5. User Interface and Its Components
    5.1 About Page
    5.2 Creator’s Page
    5.3 Future Prospect Page
    5.4 UT/UX Design with color and font palettes
7. Future Scope
8. Conclusion
9. References

ProEDA

This project is named Professional Exploratory Data Analysis (ProEDA), a dashboard of tools and options that provide an easy and enjoyable experience of seeing your data come to life. It is very user friendly, and allows you to choose exactly the data variables and trends you want to analyse, whether it is the break up of how you spend your day doing different activities in a simple pie chart, or whether it is the correlation between the age of a patient and his blood pressure in a complex scatter plot.

Other than the charting graph it also features extra tools to help the users such as notes, feedback mechanism and themes.

The user can also see the guide page which shows how to use the web-app in the about page sections. And there only the other informations of the project are also given.

This project is done as part of the credited paper in the VI semester of the Bachelors of Computer Applications course in Sri Sathya Sai Institute of HIgher Learning, Muddenahalli
campus in collaboration with:

- Sai Nath Reddy, 184402, III BCA
- B. SaiKrishna Balaji, 184407, III BCA
- Deepam Rai, 184413, III BCA

Technologies Deployed

1. **Vue JS :**

   Vue (pronounced **view**) is a progressive front-end framework based on the JavaScript Language. It is open source, and is one of the fastest growing technologies that is becoming popular for developing interactive, beautiful and responsive apps. It is a model-view-view-model (MVVM architecture) framework that is mainly used to build single page applications and futuristic user interfaces.

   It was created by Evan You, a senior developer working at Google on Angular, another front-end framework and a product of Google. He wanted to extract the features that he liked in Angular, but to make a more lightweight but feature intensive framework. The first code commit of the project was made in July 2013, in its GitHub repository. It was then released as a public product the following February.

   In Vue, everything is regarded as a component. It makes the code really easy to scale, understand or replicate because of its inherent modularity. Each component comes bundled with three facets:

   - HTML – the <template> tag
   - JS – the <script> tag
   - CSS – the <style> tag

   Thus, each component is independent, and just has to be embedded in a parent component or such to be used on the page. This doesn't mean the flow of data between components is hindered. That is smoothly carried out by features like event emitting(Child to parent data flow), and props for passing data(Parent to Child data flow).

   Advantages:

   - Simplicity – The purpose behind Vue was to make Component based web development as easy as possible for the devs, and this was achieved. Vue is also very optimized performance wise as the components have very small overheads due to the bundling up of HTML, JS and CSS into one file.

- Easy Integration – Vue code can be easily integrated into other technologies like Angular or React. Thus, it can be used to improve existing web apps very easily. Also, as it is based on the MVVM architecture, it can easily handle HTML blocks too.

- Community Support - The support for the platform is impressive. For instance, in 2015, every query on the official platform was answered. Similarly, more than 1,400 issues were resolved on GitHub with an average time of less than 13 hours. In 2018, the support continued to impress as every query was diligently answered. In fact, more than 6,200 problems were resolved with an average resolution time of only six hours. To support the community, there are consistent release cycles of updated information. In addition, the community continues to grow and evolve with the backend support of developers.

Drawbacks:

Safe to say, there are a few drawbacks too. For example, the development team has very little support from corporates for the last few years, so Vue has been widely used for personal or small scale projects only. Stability has also been a constant issue since its inception. But this is manageable for small scale applications, as the issues can be resolved quickly.

We chose Vue because of its simplicity. The learning curve of the framework was comparatively flatter and it also suited the scale of our application, while providing the benefits of a component based framework.

2. **Node JS**

   Node JS is an asynchronous, event-driven, server side platform, based on Google Chrome's Javascript R1. Select the Dataset from the machine. When the "Choose your dataset" option is clicked, it will open up a file explorer to select the .csv file from the computer. This is then loaded to the browser.
   182. Click the "Start Analysis" button. This will then in the background emit an event that sends this file to the csv file parser (Papa Parse JS library). The file will be parsed, aking the delimiter to be commas, and the first row to be the variables. The parseddata will be then sent back as an Array of arrays, with each sub array representing a row of the dataset. The data value can be accessed by using the row number, and the index of that particular variable as the column number.

3. Choose the required Variables. Once the parsed data reaches the app, it will segregate the variables into numerical ones and categorical ones, and they will be displayed appropriately. Two numerical variables can be selected at once(for the scatter plot), but only one factor variable can be selected at once. According to the variables chosen, the graph options will be enabled or disabled.

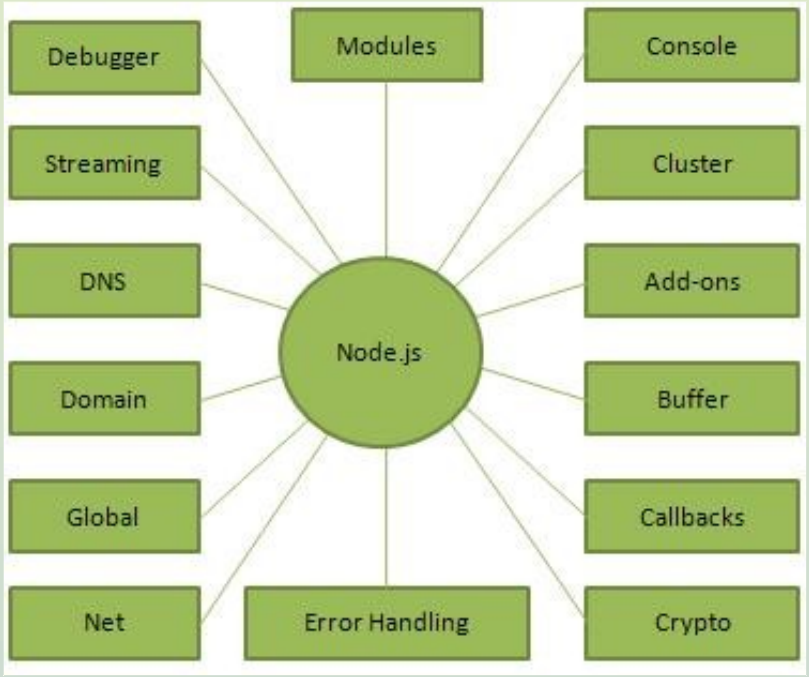
● If one numerical variable is selected, the line, bar, column and area charts will be enabled.

● If one factor variable is selected, the pie and histogram chart options will be enabled.

● If two numerical variables are selected, the factor variable selection is not considered and the scatter plot option will be enabled.untime, that can be used to build heavily scalable applications easily. It is one of the main components of most Javascript based full web stacks. It is largely popular due to the availability of extensive and exhaustive Javascript modules that simplify the building of the connecting platform between the server and the frontend client.

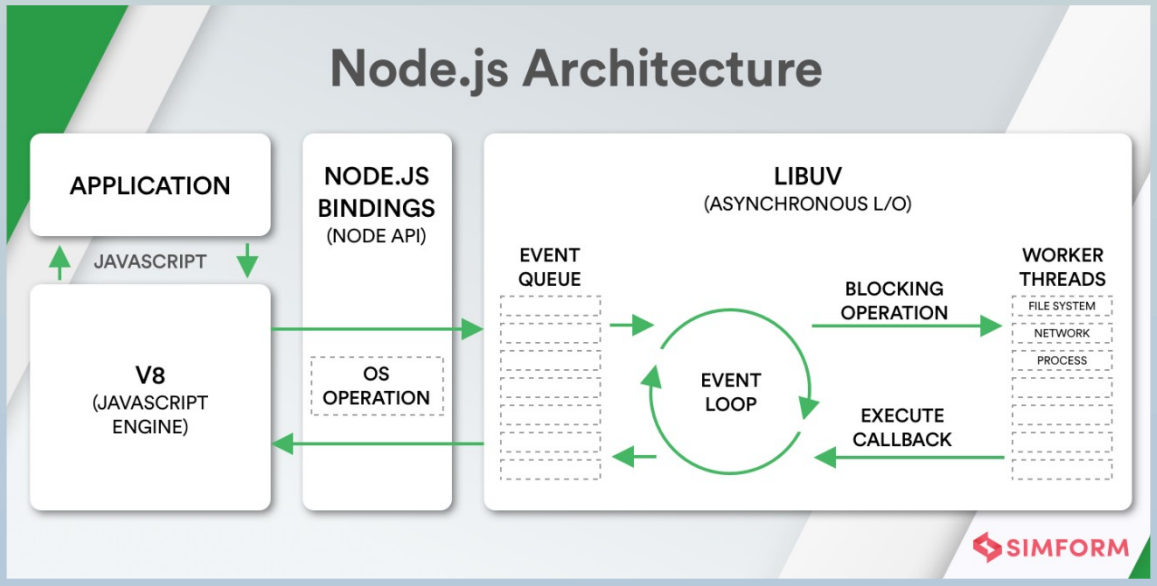
It is blazing fast because it is asynchronous. It never waits for an API call to return the requested data. It executes it and moves on to the next call, using its callback system to capture the return of the previous call. It is also heavily optimized because it uses a single threaded event based system. Thus no deadlocks can be caused in the OS, unlike many other server platforms which use the multithread based concurrency system.

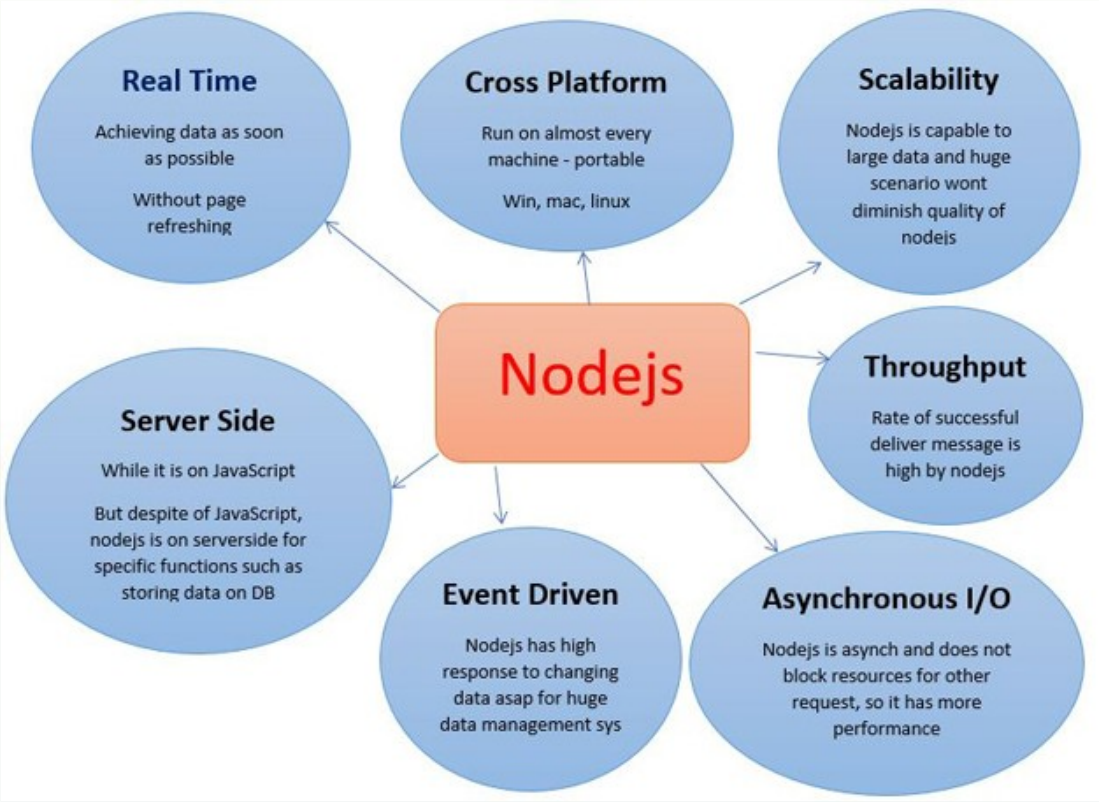
Node JS is used extensively on many commercial and heavy corporate projects such as, eBay, GoDaddy, Yahoo!, Uber, PayPal, to name a few. Its reliability and widespread popularity can be gauged based on its usage in these giants’ applications.



One of the biggest offerings of Node Js is its package management system, **npm**. It is one of the largest package library managers out there, and there are millions of absolutely free packages offered with regular updates and a slew of features. It is a main staple of full stack developers as it allows easy integration of libraries and other tools into the app.

One of the biggest offerings of Node Js is its package management system, **npm**. It is one of the largest package library managers out there, and there are millions of absolutely free packages offered with regular updates and a slew of features. It is a main staple of full stack developers as it allows easy integration of libraries and other tools into the app.

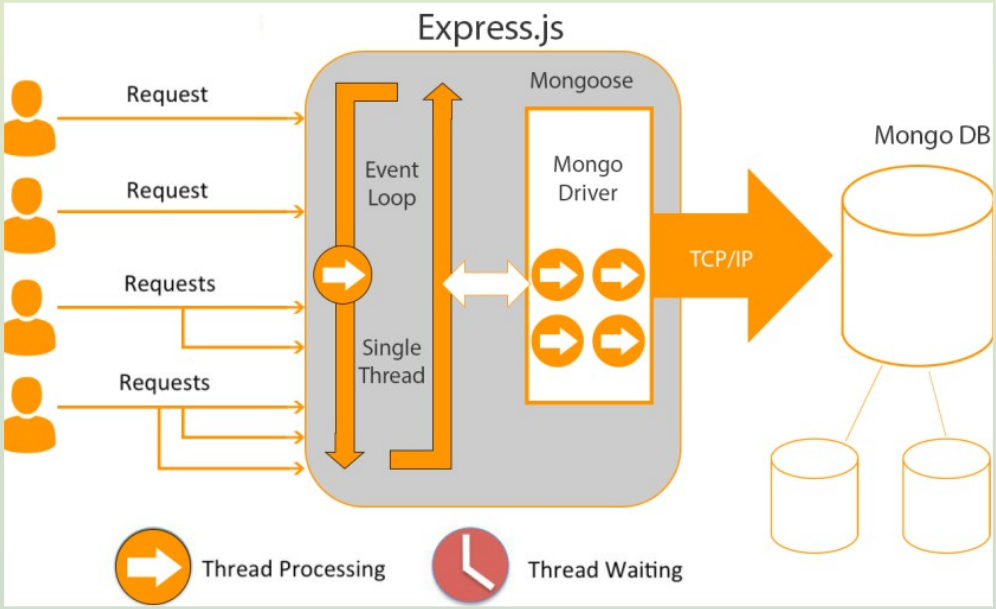




3. **Express JS**

   Express JS is a minimal and flexible Node JS application that provides robust features for web and app development. It is a very thin layer that offers fundamental development libraries without obscuring the features of Node JS itself. It is also very light, thus heavily improving performance. It is used to build APIs and offers easily useable middleware to the full stack.

   These days, wherever Node JS is used as a server side platform, Express JS is often used as the middleware. It is really easy to work with Express and to make it handle the connection between the Client side requests and the server side responses.



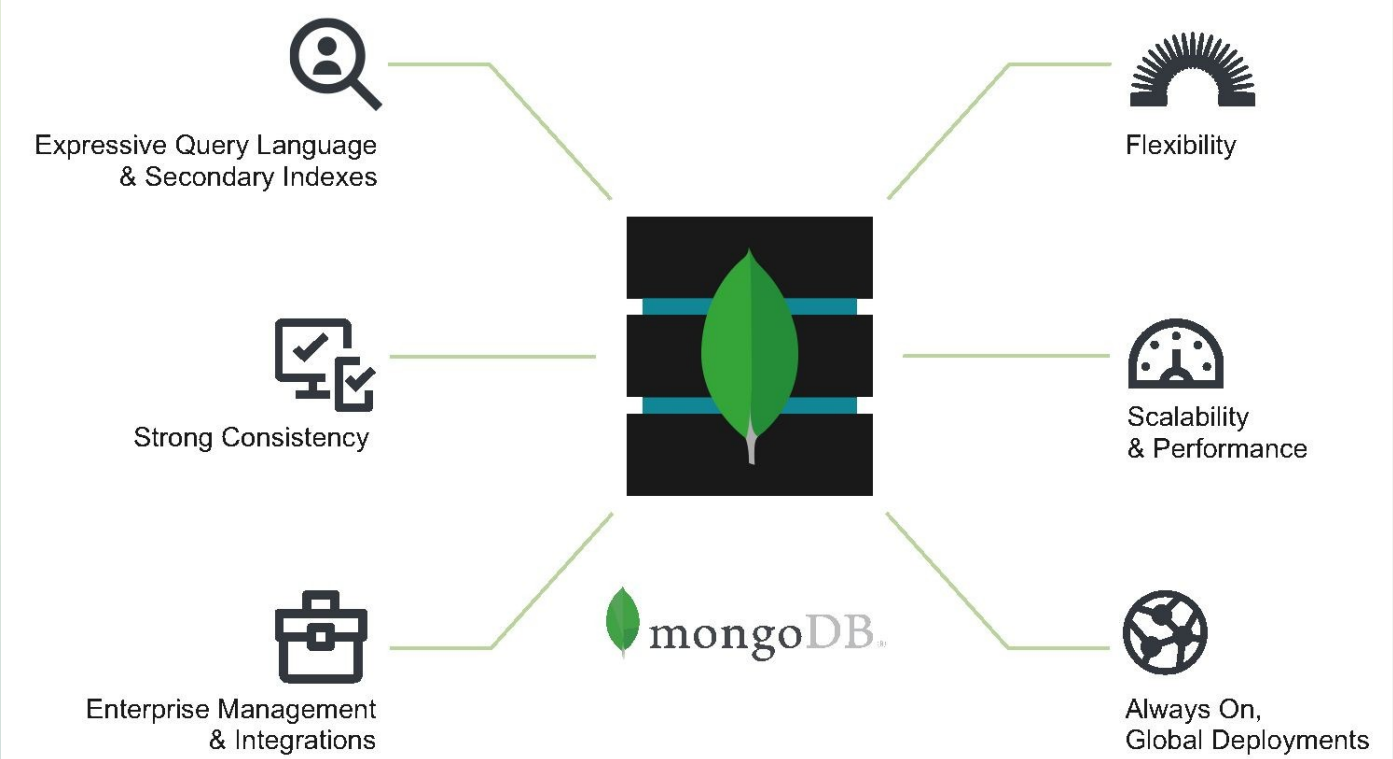
4. **MongoDB**
   MongoDB is a general purpose, document-based, distributed, NoSQL database built for modern application developers. These days, most developers are used to thinking in terms of objects, its properties and data. This DB stores its records in the collections in a similar manner, using JSON. This makes it so easy and comfortable for programmers to send and receive the data from the DB.

   A NoSQL database(*not SQL, not only SQL, etc.*) is a non relational database, meaning the data isn't stored in relations, a.k.a rows/columns format. This type of database gives maximum flexibility to the developer. Developer Comfort became a higher priority than storage space since the late 2000s, due to reducing costs in storage mediums. Thus, the interest in NoSQL databases started to increase.

   People think that since there are no row column relations between the data, there is no storage of data of the relationships between them. This is wrong. They just store it in a different way. Many actually prefer NoSQL to relational databases in terms of relationships, because the data doesn't have to be split in between the tables like primary and secondary keys.

   Another advantage is the feature of cloud sync. The MongoDB Atlas cloud database service offers maximum comfort to developers, as there doesn't have to be any local database or server service running to connect to the DB. The connection happens through a URL provided by the cloud service according to

your IP and it is then used by Express JS and Node JS to connect to the URLString.



5. **Google Charts API**
   The main mechanism of the whole project, the charting of the selected data and the graph type is done via the google charts API. The Google charts API is a rich collection of HTML 5 Canvas based charting tools that are really easy to use and modify to need. They take in data in a very flexible manner. The key feature that sets this API apart from the countless other Charting APIs are the sheer number of chart types supported. There is even support for making your own custom chart type.

   The charts are responsive by nature and come with animations bundled in stock. The numerous chart options make customizing the charts output down to the exact details that are required for the exact application very easy. Another key feature of the API is the level of the documentation. The forums and the community around it is very active and supportive for newbie users.
   Though this API could've been integrated with the **Vue** frontend using just vanilla Javascript, there was a vue wrapper for the charts API found on GitHub

called vue-google-charts, which allowed the Charts to be mounted to the template section of the parent component in the form of a child component. The `chartData` and the `chartOptions` could then be passed to the component in the form of props and its own native data in the `data()` section of the `<script>` part of the parent component.

6. **Papa Parse**
   Papa parse is a JS library that allows easy parsing of CSV files into formats required for data analysis by the app, formats that are more conducive for the devs to extract the data from the dataset and analyse it.

   Initially, the parsing had been implemented from ground up natively, but it wasn't proper and done to a level enough to integrate with the Google Charts API. Thus, in search of a good parser, this library was found. It gives numerous options to customize the options in parsing, such as reading the first row as the table headers(as usually the first column contains the data variables), selecting the delimiter used in the dataset, whether the output parsed data should be an array of arrays or JSON objects. These options really helped to bring the parsed data to a level of easily importing into the charts API.

   This too could've been implemented as vanilla Javascript but was done by a vue wrapper found on GitHub called vue-papa-parse.

Many of these technologies that were used were gibberish to us when we first started the project development. Slowly through the help of the internet community, the expertise shared through endless tutorials in the form of YouTube videos or Blog posts, and our own interest to make this project better with newer technologies, we were able to integrate these technologies fairly well. That said, there are many places that we can use these tools to further improve our project.

The exposure that we gained from using these full stack tools to program this app is invaluable as, though carried out in a very small scale and with questionable formal software engineering practices, this is industry-related and industry-relevant experience that can be carried forward and onward.


Vue.js


Google
Developers
Charts


node
JS

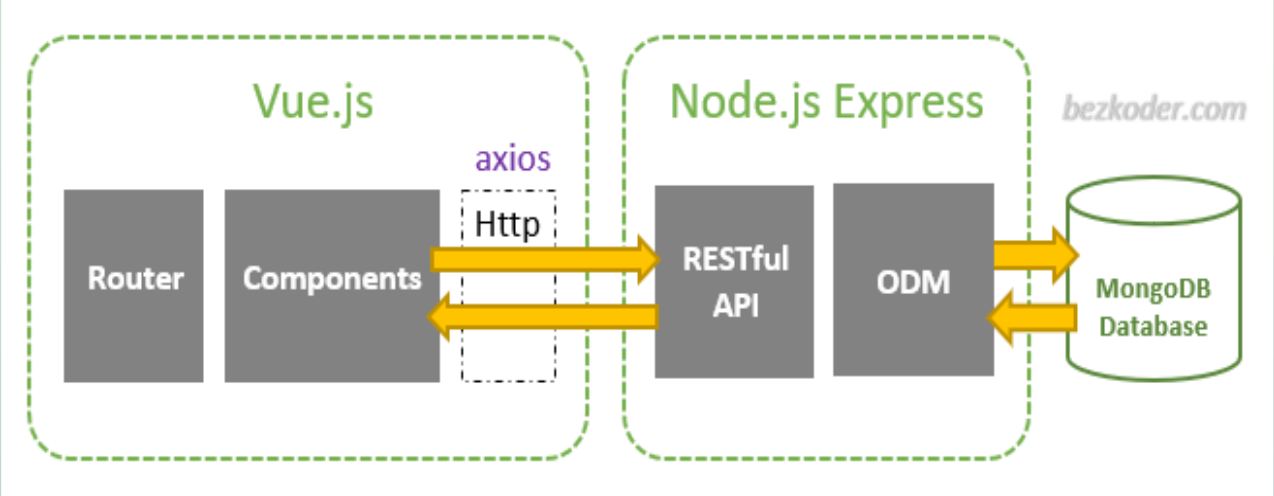

Express JS


mongoDB.


Papa Parse

Thus, all the technologies put together, the basic four – Vue JS, Express JS, Node JS, and MongoDB – form what is commonly called, the VENM stack. This contains a frontend framework, a backend Database, a server side platform, and a middleware connecting API. Some other popular stacks used are MERN(React JS instead of Vue), MEAN(Angular JS instead of Vue), and LAMP(Linux, Apache, MySQL, Perl/PHP/Python).

One of the major advantages of the VENM, MERN or MEAN stacks are that they use Javascript end-to-end in the stack. This makes life extremely comfortable for web developers as they do not have to struggle with learning and switching to and from different languages, and with them, their thinking processes and syntax issues.



The Development Platform (Pics for all)

The Software Applications that have aid the project for teh efficient develpopment of our MathEDA are as follows :-

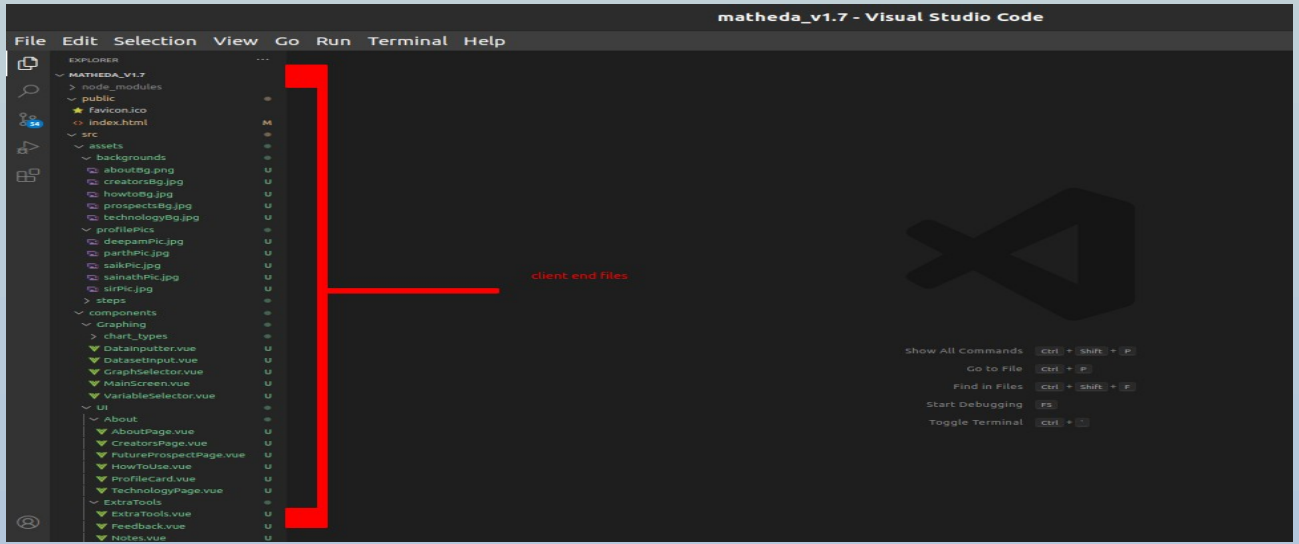
Terminal or Bash shell :

Terminal or Bash shell is ubuntu's Command line interface where-

- Open the Application folder in the terminal
- Install all the required supports and modules to run the json file present
- compile and create local server

Visual Studio Code:

It is a light weight code editor built by Microsoft. Our whole project was structured using VScode. It comes with essential extensions for MERN files, which helps in code formatting, syntax highlighting, quick-fixes, code suggestions and code autocomplete. It has also helped us in integrating client and server side effectively.



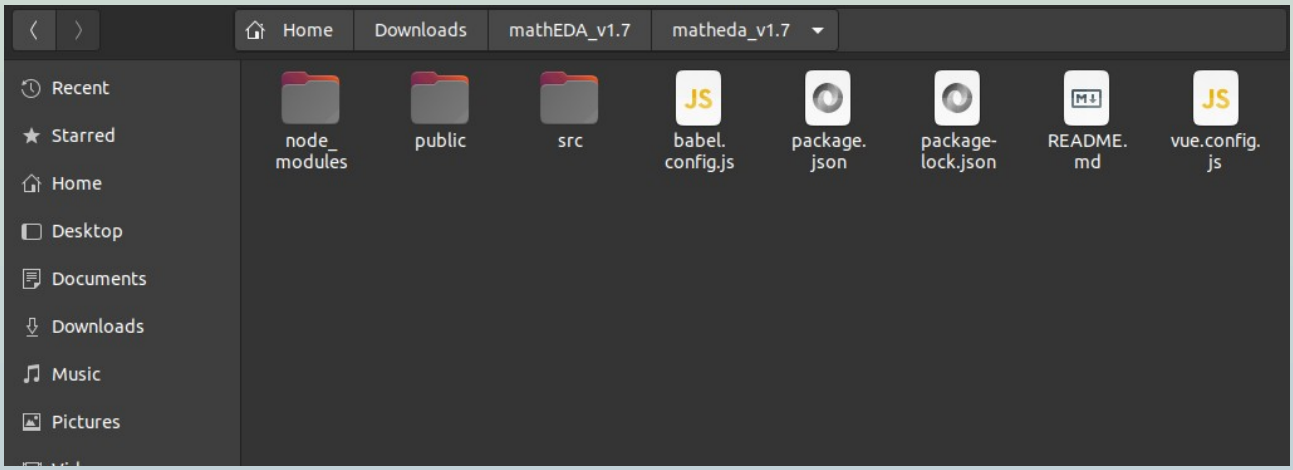
Google Chrome

Google Chrome is a cross-platform web browser developed by Google. It also has the same V8 engine, which effectively runs and debug our React view pages. Our client side code has been fully tested using Google Chrome browser.

Hardware Configurations :-

OS – Ubuntu 20.04 Focal Fossa
OS Type – 64bit
Ram – 4GB
Processor – Intel i3

Ubuntu was used as it was easy to use npm and the bash terminal easily for installing packages and running the development server for local testing.



WORKFLOW

1. Select the Dataset from the machine. When the “Choose your dataset” option is clicked, it will open up a file explorer to select the .csv file from the computer. This is then loaded to the browser.

2. Click the “Start Analysis” button. This will then in the background emit an event thatsends this file to the csv file parser (Papa Parse JS library). The file will be parsed, aking the delimiter to be commas, and the first row to be the variables. The parsed data will be then sent back as an Array of arrays, with each sub array representing a row of the dataset. The data value can be accessed by using the row number, and the index of that particular variable as the column number.

3. Choose the required Variables. Once the parsed data reaches the app, it will segregate the variables into numerical ones and categorical ones, and they will be displayed appropriately. Two numerical variables can be selected at once(for the scatter plot), but only one factor variable can be selected at once. According to the variables chosen, the graph options will be enabled or disabled.

- If one numerical variable is selected, the line, bar, column and area charts will be enabled.
- If one factor variable is selected, the pie and histogram chart options will be enabled.
- If two numerical variables are selected, the factor variable selection is not considered and the scatter plot option will be enabled.

Snapshot of summary of all Steps :

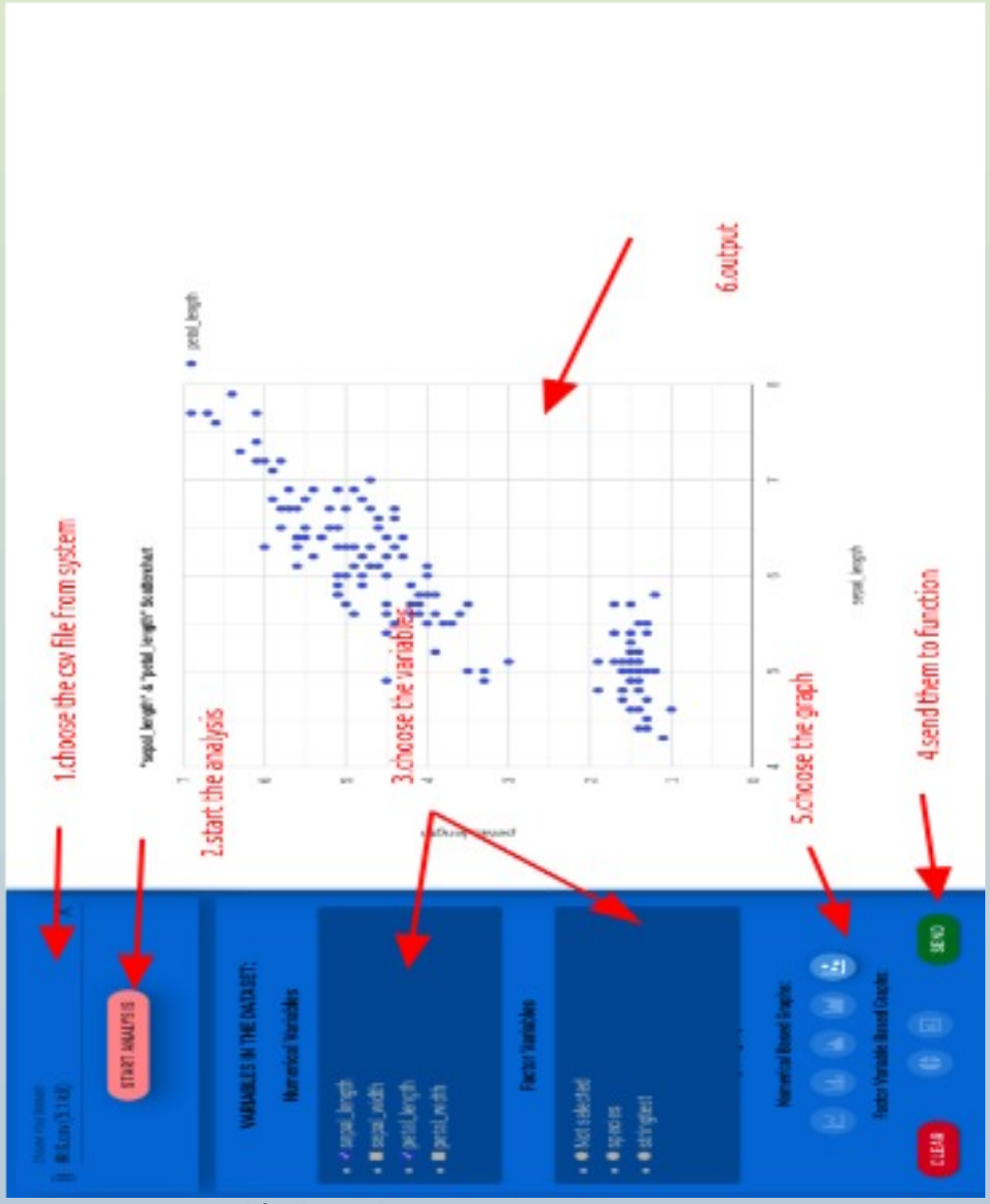


Figure 1: Steps to proceed for the graph generator

User Interface And Its Components

About Page:

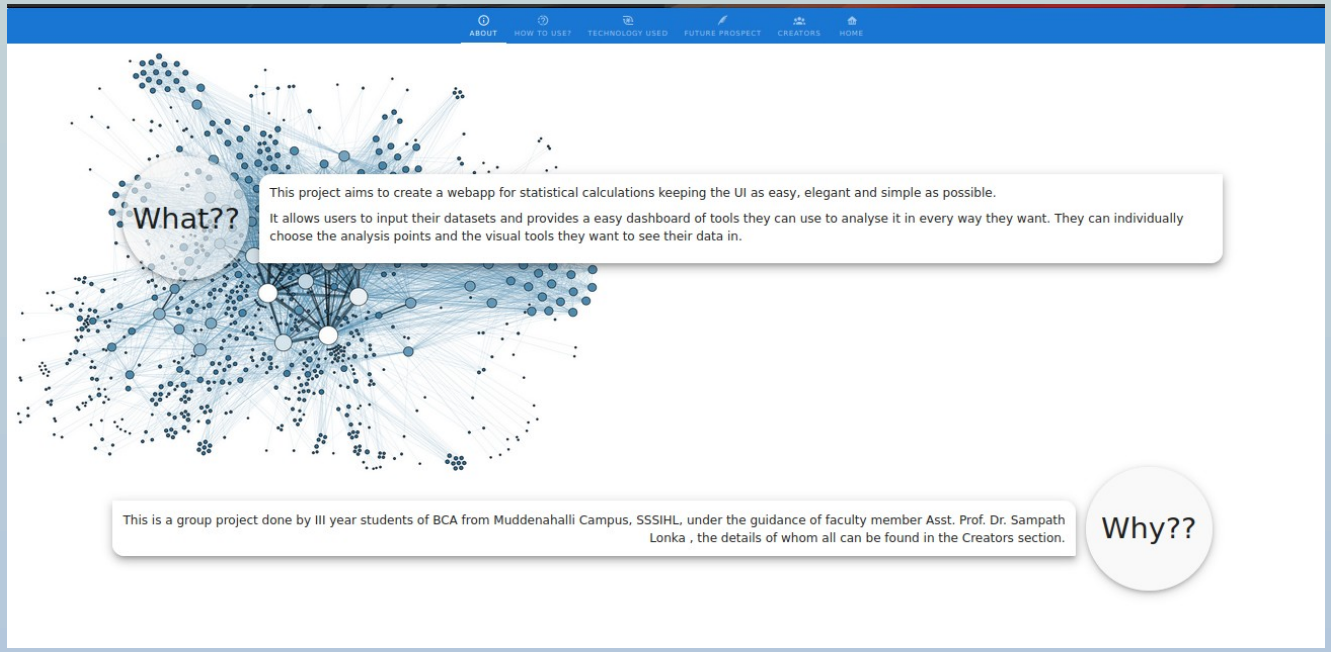
An About **page** is a special web **page** on a site where your readers/visitors learn more about you and what you do. And here fulfilling the same reason, we create a component which will render as a tab in view About.vue and tells about the project and vision.

Under the class, aboutPage, created two acrs which will show the desired text to the user. With rightly margined to both side and a suitable background using vuetify. For the subheadings, <v-avatar> tag is used which will come as circle on both extreme of either of text card.

<-- Basicallyh major items are flex div contaning an avataar and text Just that the flex directions are changing alternatively -->

```
<div class="aboutPage">
  <div class="leftPslabs">
    <v-avatar class="avatars" size="250">
        What??
    </v-avatar>
    <div >
      <v-card class="rightText" elevation="10" shaped>
        <p>This project aims to create a webapp for statistical calculations keeping the UI as easy, elegant and simple as possible.</p>
        <p>
        It allows users to input their datasets and provides a easy dashboard of tools they can use to analyse it in every way they want. They can individually choose the analysis point
      </v-card>
      </div>
  </div>

  <div class="rightPslabs">
    <v-card elevation="10"  :class="`rounded-tr-xl rounded-bl-xl`">
      <div class="leftText">
        This is a group project done by III year students of BCA from Muddenahalli Campus, SSSIHL, under the guidance of faculty member Asst. Prof. Dr. Sampath Lonka , the details of wh
      </div>
    </v-card>
    <v-avatar class="avatars" size="250">
        Why??
    </v-avatar>
  </div>
</div>
```



Creator's Page :

Creator profiles can designate their preferred method of contact on their profile (including call, text, and email), and have more flexibility on what they choose to include as a CTA on their **pages**. Regarding this, There is one vue page is create which creates "Creator Page" for the project under the class CreatorPage.

There are two Profile cards depending on the designation:

- Teacher Profile card
- Student Profile card

Using the v-Avatar component, the tile prop, we can create a sleek hard-lin ed profile card which will show details containing name and their emails. Profile card will be imported from

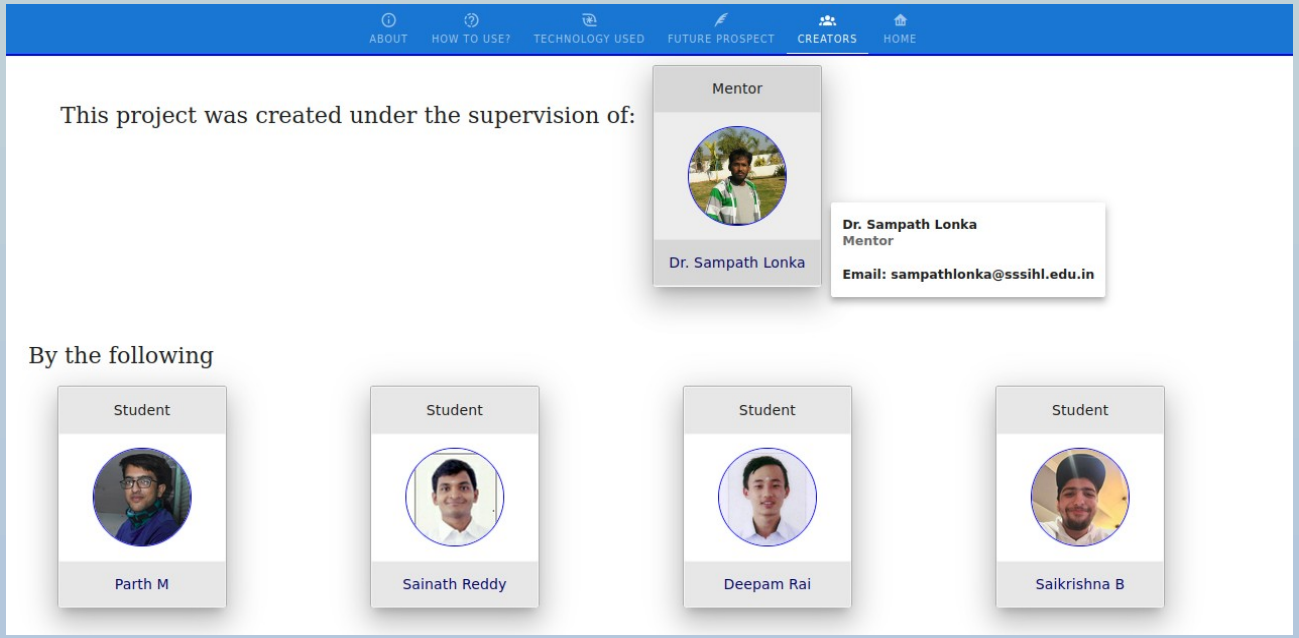
module "/home/prudentemills2502/Downloads/mathEDA_v1.7/matheda_v1.7/src/components/UI/About/ProfileCard.vue"

and the Pictures for all admins will be retireve from assessts/profilepic.

```
      <v-col>
        <br><br>
        <h1 style="text-align:right;">This project was created under the supervision of:</h1>
      </v-col>
      <v-col>
        <!-- mentor profile card -->
        <ProfileCard
          :pname="mentor.name"
          :ptitle="'Mentor'"
          :pemail="mentor.email"
          :psrc="mentor.pic"
        />
      </v-col>
    </v-row>

    <!-- student profile cards -->
    <v-row style="margin-top: 4%;">
      <h1 style="margin-left:3%;">By the following</h1></v-row>
    <v-row class="studentRow">
      <v-col v-for="item in students" :key="item.index">
        <ProfileCard
          :pname= "item.name"
          :ptitle="'Student'"
          :psrc= item.pic
          :pemail= "item.email" />
      </v-col>
    </v-row>
  </v-container>
</template>

<script>
import ProfileCard from './ProfileCard.vue'
```

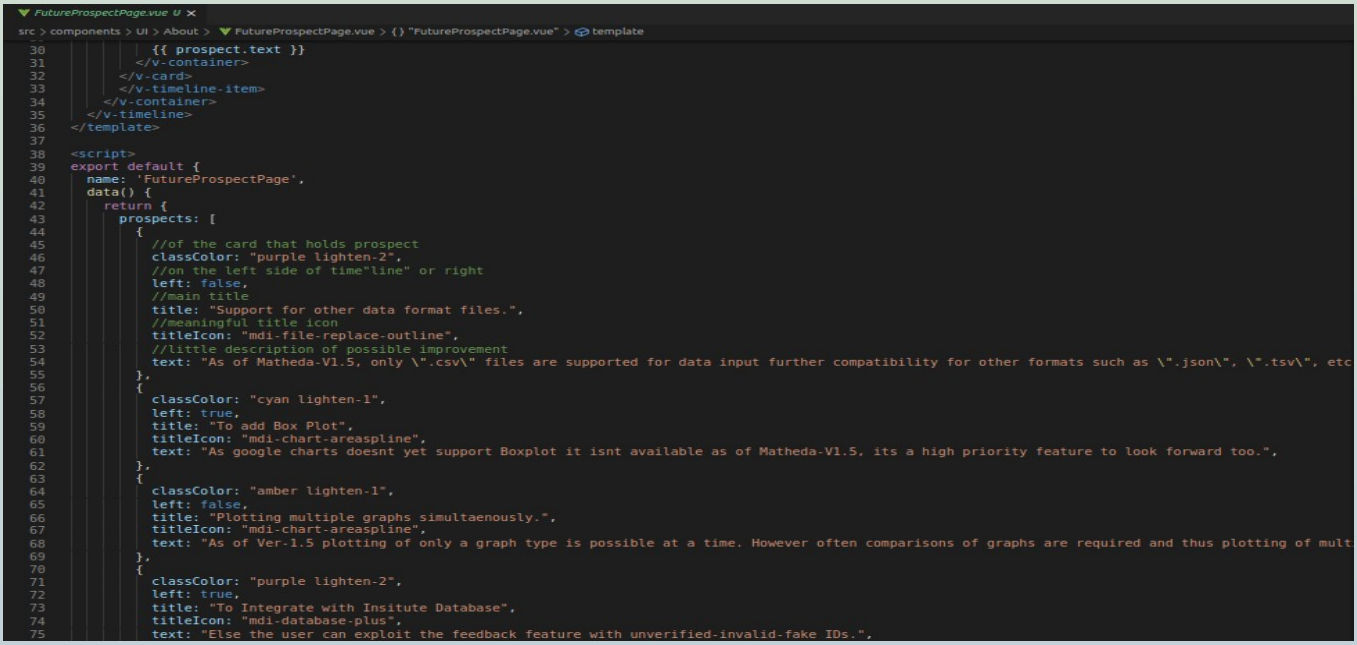


Future prospects Page:

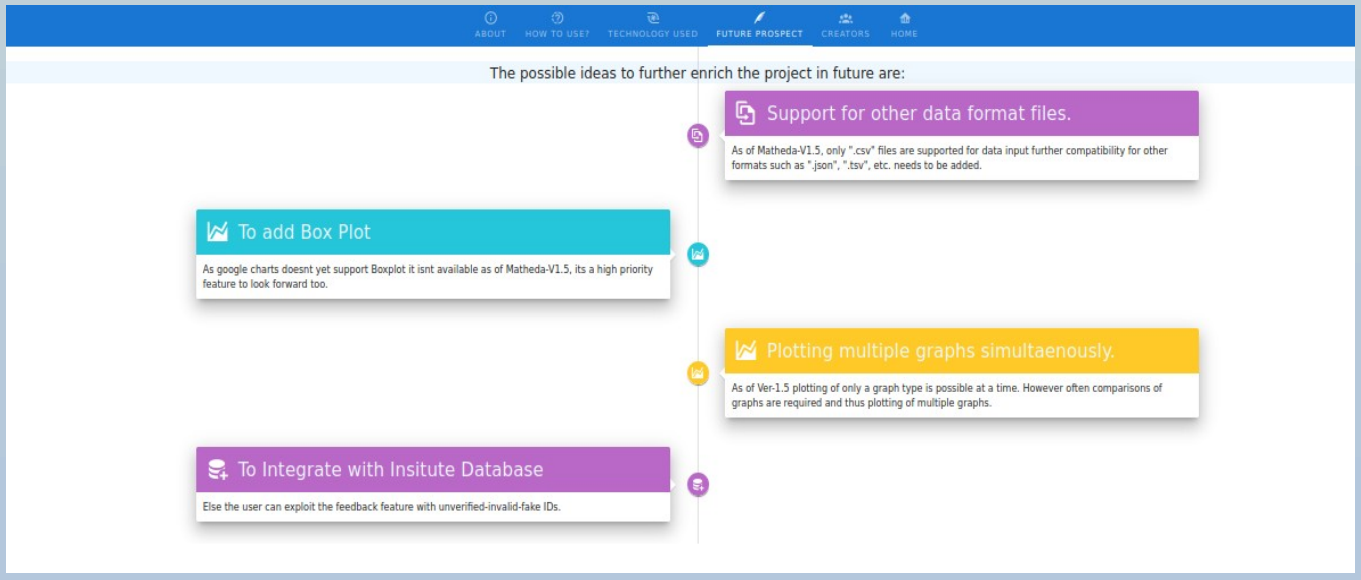
A **prospect** is still a way of looking ahead and expecting good things. This page gives the possible improvements on the project in the future. Of course the list is not exhaustive thus this page only features few of the prospects.

A container contains the prospect.txt file which has data to show and it will be portray in a card. The points under prospecr are shown in the cards either side of a middle line generated as a tree.

There is colour selection in all cards in order to distinguish from each other.



```
FutureProspectPage.vue U
src > components > UI > About > FutureProspectPage.vue > {} "FutureProspectPage.vue" > template
30            {{ prospect.text }}
31          </v-container>
32        </v-card>
33        </v-timeline-item>
34      </v-container>
35    </v-timeline>
36  </template>
37
38  <script>
39  export default {
40    name: 'FutureProspectPage',
41    data() {
42      return {
43        prospects: [
44          {
45            //of the card that holds prospect
46            classColor: "purple lighten-2",
47            //on the left side of time"line" or right
48            left: false,
49            //main title
50            title: "Support for other data format files.",
51            //meaningful title icon
52            titleIcon: "mdi-file-replace-outline",
53            //little description of possible improvement
54            text: "As of Matheda-V1.5, only \".csv\" files are supported for data input further compatibility for other formats such as \".json\", \".tsv\", etc
55          },
56          {
57            classColor: "cyan lighten-1",
58            left: true,
59            title: "To add Box Plot",
60            titleIcon: "mdi-chart-areaspline",
61            text: "As google charts doesnt yet support Boxplot it isnt available as of Matheda-V1.5, its a high priority feature to look forward too.",
62          },
63          {
64            classColor: "amber lighten-1",
65            left: false,
66            title: "Plotting multiple graphs simultaenously.",
67            titleIcon: "mdi-chart-areaspline",
68            text: "As of Ver-1.5 plotting of only a graph type is possible at a time. However often comparisons of graphs are required and thus plotting of mult
69          },
70          {
71            classColor: "purple lighten-2",
72            left: true,
73            title: "To Integrate with Insitute Database",
74            titleIcon: "mdi-database-plus",
75            text: "Else the user can exploit the feedback feature with unverified-invalid-fake IDs.",
```



UI/UX design with color and font palettes:

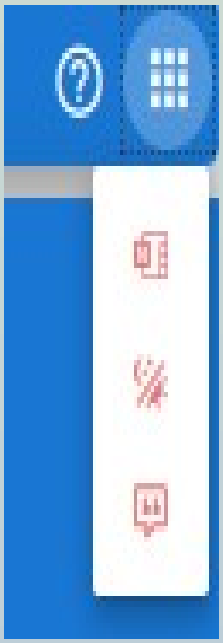
The web-app UI is built around simple fonts and colour palettes. They are chosen as browser safe and hence presents no complications of being rendered falsely on some browsers while functioning properly on others.

The basic colour palettes are controlled via themes which affects the whole of the webpage, and which is accessed through the extra tools section.

There are a list of colours given to choose from in the theme section namely: sky, greeno, pinko, mango, ocean, dark theme. User just have to select one and click on it to see the desired colour flourish throughout the webpage.

```
<script>
import ExtraTools from './ExtraTools/ExtraTools.vue'

export default {
  name: 'TopDashboard',
  components: {
    ExtraTools,
  },
  props: ['ptheme', 'version'],
  data() {
    return {
      drawer: false, //to control the side-bar display
    }
  },
  methods: {
    changeTheme( newTheme) {
      //to change the theme
      //event received from ExtraTools<Themes;
      //emit to main Home.vue
      this.$emit( 'changeTheme', newTheme);
    }
  }
}
```



Future Prospects

A road never ends but maintained and decorated for a long-term use. Similarly there is exhaustive amount of scope of addtional improvements to the current application. To Increase the potential of the application based on the User Interface and other growing stuff in the field of EDA and Graphs, here our some visions to fulill yet-

- User Interface can be more user friendly with useful options like Quotes, Direct links to references and some fancy options like dark theme etc. With the suitable Background images to it.

- There is always possiblity of adding an login system to an application with the stable tech and can be further useful in our own campus to provide aid in both the deopartments

- Side toggleing tool bar to provide more options on less space to show more data on screen.

- Summarization of One graph with all the variables either on seperate page or sidebar

- There is always scope of adding more options of Graphs with more accurate results with more detailing if two or more variables are used.

- inclusion of support for uploading the dataset file from Google Drive or Dropbox, as these are the platforms used by many people to store their data online for access anywhere.

-Visualization can be improved in order to give the user more precision of the graph and its features.

CONCLUSION

This project isn’t without its own limitations. The UI could be further improved for easier user usage. The charting could be further optimized for more clarity and better performance with regard to big datasets. But this is a humble attempt to make a full stack JS web application that makes data visualization simpler for the user, with holistic goodness.

The project will be further developed and supported and many more better versions will be put out in the future that keeps on this basic idea of a Professional Exploratory Data Analysis (ProEDA) dashboard app.

REFERENCES

From used technologies official home pages, documentations and communities:

1. Vue Framework : https://vuejs.org/
2. Vuetify Component Library : https://vuetifyjs.com/
3. Node.js : https://nodejs.org/
4. Express : https://expressjs.com/
5. Axios Library : https://www.npmjs.com/package/axios
6. MongoDB (No-SQL Database) : https://www.mongodb.com/
7. Google-Charts : https://developers.google.com/chart
8. Papa-Parser (CSV file to JSON) : https://www.papaparse.com/
9. BodyParser (DB objects to JSON): https://www.npmjs.com/package/body-parser

From the Q/A internet sites:

1. Stackoverflow : https://stackoverflow.com/
2. Superuser :https://superuser.com/

Turorial sites:

1. youtube: https://youtube.com/
2. w3schools: https://w3schools.com/
3. freecodecamp: https://freecodecamp.org/
4. tutorials point: https://tutorialspoint.com/

Thank You!

We are forever indebted

and grateful

to our beloved Mother Sai...