BACKEND DEVELOPMENT
USING NODE-JS MONGO-DB AND EXPRESS

Application Management System, Board Infinity

A training Report

Submitted in partial fulfillment of the requirements for the award of degree of

Bachelor of Technology

Computer Science and Engineering

Submitted To

LOVELY PROFESSIONAL UNIVERSITY
PHAGWARA , PUNJAB

From 06/01/2021-07/15/2021

SUBMITTED BY

Name of Student : Sudeep Kumar
Registration Number: 11917205
Signature of Student : Sudeep Kumar

LOVELY PROFESSIONAL UNIVERSITY

Table of Contents

LOVELY
PROFESSIONAL
UNIVERSITY

Student Declaration

I, **Sudeep Kumar, Registration Number 11917205,** hereby declare that the work done by me on "**Backend Development Using Node JS Mongo DB and Express**" from **June, 2021** to **July, 2021**, is a record of original work for the partial fulfillment of the requirements for the award of the degree, **Bachelor Of Technology Computer Science and Engineering.**

Sudeep Kumar (11917205)

Sudeep Kumar

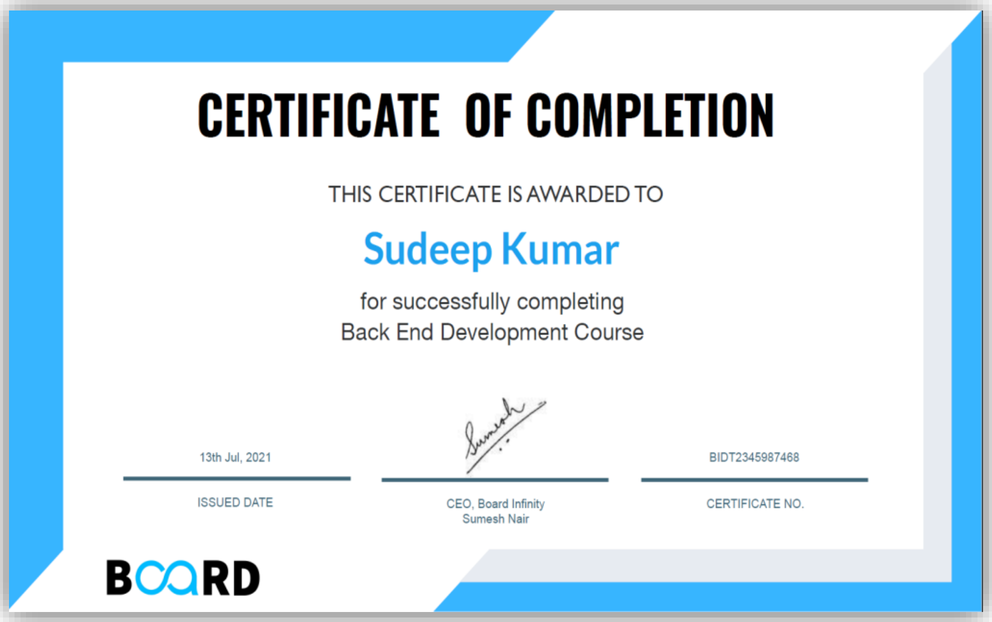
Signature of the student

Dated: 28 Sept 2021

LOVELY
PROFESSIONAL
UNIVERSITY

TRAINING CERTIFICATE

Backend Development Course
Board Infinity



CERTIFICATE OF COMPLETION

THIS CERTIFICATE IS AWARDED TO

Sudeep Kumar

for successfully completing
Back End Development Course

13th Jul, 2021
ISSUED DATE

Sumesh
CEO, Board Infinity
Sumesh Nair

BIDT2345987468
CERTIFICATE NO.

BOARD

ACKNOWLEDGEMENT

In the present world of competition there is a race of existence in which those are having will to come forward succeed. Project and training is like a bridge between theoretical and practical working. With this willing I joined this particular training. First of all, I would like to thanks the supreme power of Almighty God who is obviously the one has always guided me to work on , the one has always guided me to work on the right path of life. Without his grace this project could not become a reality. Next to him are my parents, whom I am greatly indebted for me brought up with love and encouragement to this stage. I am feeling oblige in taking an opportunity to sincerely thanks my teachers and mentors of Lovely Professional University who guided me towards doing this "Summer Training on Backend Development Course" which helped me to know the different aspects of technology like HTML, CSS , JavaScript , Node JS ,Mongo DB etc. I am highly obliged in taking the opportunity to thanks the "Board Infinity , Navi Mumbai , India" and to Mr. Nanda Kishore who helped me and guided me to learn more about the new technologies, and also taught me to integrate Database like MONGO DB and MY-SQL and helped me to do something realistic with the Data .I also want to thank the platform of Board Infinity which help me to learn such technologies.

At last but not the least I want to thanks my friends, classmates and colleagues who help me in completing this Summer Training in Backend Development with Node JS and Mongo DB and for encouraging me for do this types of useful project to boost my knowledge. I have no valuable words to express my thanks but my heart is full of all the favors received from every person.

Sudeep Kumar

11917205

LOVELY PROFESSIONAL UNIVERSITY

LIST OF TABLES

LIST OF FIGURES

LOVELY PROFESSIONAL UNIVERSITY

LIST OF ABBREVIATIONS

S . No	Abbreviations	Full Form
1.	HTML	Hyper Text Markup Language
2.	CSS	Cascading Style Sheet
3.	JS	JavaScript
4.	DB	Data Base
5.	CRUD	Create, Read, Update, Delete
6.	CDN	Content Delivery Network
7.	REST	Representational State Transfer
8.	API	Application Programming Interface
9.	SQL	Structured Query Language

CHAPTER – 1

INTRODUCTION

As we are living in the developing world and the one of the most important aspect now a days is technology. A country who is more advanced in technology considered better as compared to those who are little less developed or who are currently a developing country. In this developing world there are bunch and bunch of technologies which seems countless but on the other hand very much important also as most of the operations now a days depends on these technologies. One of these bunch of technologies is the Development technologies, where the developers used to develop desktop applications, websites ,software, API and many more things using some computer tools and languages like C++, C, Java, Python, R ,HTML ,CSS , JavaScript, Node JS, Mongo DB, React and many more like these.

In current world , development is one of the emerging and one of the most important technology , without development any thing is incomplete. As we can see today, websites and mobile application are that much important that it become the part of our daily life. We use websites and app in almost each and every domain of life like Banking, Cinemas, Business, Movies etc. These applications made our life easy and also saves our time. We cannot imagine our life without these in current era or world. Hence it is somewhere very important for a student that he or she should have some good knowledge of Some of these Technologies.

Hence ,This project is also based on one of such technologies. It is a project based on User Application Management System . It comprises of HTML, CSS, JavaScript, Node JS, Mongo DB , Express .It is just like a examination registration portal where candidate fill the application and get the details in the acknowledgement page as Application acknowledgement . It will represent some glimpse of how things actually work using a Demo of application system where the user come and signup and can fill the application form and able to see the preview and download the print of the preview screen . The user can also sign in using the Registration Number generated while signup and the password created by the user and can update the data entered earlier and can also be able to change the sign in details. All the data are stored in the Database which is using Mongo DB Atlas. Mongo DB is the database which used to store the information in the JSON format which is basically just a object of a JavaScript program and called as JavaScript Object Notation(JSON).

In this project there is a server build in by using Node JS which have some endpoint from where it stores the information into the database or retrieving information from the database. It Will also implement Mongoose to define schema and do operation with the database. It consists of some user interfaces for the user like Login, Signup, Application , Application Preview, Edit Application , Forgot Application, Change Password which is made by using the Concepts of HTML , CSS and JavaScript. User just have to go to the website and do the operations and each backend operations will be handled by the Node JS.

This project will also use some new and handy technologies like Express , REST API etc , which is very much useful. Express generally used to put the request to the Server and on the other hand Rest API will give us the information about the data coming from that request from the server side. It will give use the information like which type of request it is, the status code, the types of information coming , error if there is any and some other important information.

This project uses the Platform Visual Studio Code environment to do all the required stuffs like coding, logic implementation etc. There are many other platform to go with but Visual Studio Code is very much famous as well as preferable so it take use of Visual Studio Code. This project is the implementation of all those stuffs which we should know for backend development as a beginner and can be modified for the better experience in the coming future.

So, its just the intro of Why, What , and How !! about the project so that the reader get some idea what it is all about and how thing will be going to work . The actual implementation and logics are all mentioned in upcoming details of this project.

TECHNOLOGY LEARNT

This project uses various technologies. Since this project is based on Web, So mostly it used web technologies
The technologies used are mention below:

- **HTML (Hyper Text Markup Language)** : It is a markup language used to design web pages and can be understood by HTTP or HTTPS Protocols. It consists of number of tags which is used to give the basic structure to the web pages. It is also very much demanding in the real world as in this modern world, we can say most of the things are based on internet and websites and it is a building block of any websites. Facebook, Twitter, Amazon are some of the big websites using this technology. In this project it is again used as building block of our webpages.

- **CSS (Cascading Style Sheets) :** It is a language used in websites to style the webpages and give them attractive looks. It is used to give every element of style we can think of to the webpages like animation, transition , designing , etc…. All the design in this project is created by using this CSS . In real world again this is also very useful. You can go to any website and you see you will find it attractive , there are various transition effect and also various animations . You see different styled button, links , Nav-bars , etc. This all is done with the help of the CSS.

- **JavaScript :** It is again a language or more correctly a scripting language . which is used to manipulate the webpages dynamically. It is used to add functionality to different element of the webpages like buttons , links etc. It is used to make our website work and function, each and every thing our website do is done by JavaScript and its used concept of DOM (Document Object Module) to do all sorts of things. If we don't use this our webpages is just like a static image and we will not be able to do any operation in it. Our whole project implement this JavaScript for various functionalities of buttons, links or to do any request form front end to backend . In real world again it is very useful to make things work in the web technologies.

- **Node JS :** It is one of the newly originated technologies which is in use from recent decades. It help us to integrate the JavaScript into the Backend Development. In previous time we used different technologies like php , .NET (Dot NET) etc. to create backend but Node JS help us to do it by using just the JavaScript and it become very useful because we donot need to learn various languages for it.Node JS is the bundle of we can say the library of the JavaScript oriented logics which kept on single place and are used to do backend development . In this project Node JS is used to make the backend and the server side logic. And now a days in real world its demand is increasing because it is very useful as it give ease to the developer. And also it provide a complete node module to do different tasks which seems to be very difficult if applied manually and all these things are managed by NPM(node package manager).

- **Express :** Express is just a module come with Node JS which help us to do request in a easy way and also in an efficient way. Without using express we can also do stuffs but that become little lengthy and complex as we have to take care of different things. Express give us ease and provide some inbuilt functions to do all the same tasks in a very simple manner. Hence it is very effective and developer friendly and it also saves our time. In Current world wherever the Node JS is being used express is the first choice of the developer as it give them that ease to do things and manipulate some of the complex things while developing.

- **Mongo DB :** It is a technology which deals with the database. It is a database management system which help use to store data in a particular format in their cloud storage and let us do whatever operation we want to do with the stored data. Some of the basic operations are Create, Read, Update, Delete called as CRUD . It is also called No SQL database since it donot implement the relational database logics means it donot based on table creation . It Is a database Based on JSON logic, means it stored all the information in the list of JSON Objects and let us use those data. It Provide us a cloud space called Mongo DB Atlas where we can store the data. In this project all the data are stored in Mongo DB database which can be seen using Mongo DB Atlas. In the real world it is a emerging Database system and now a days many companies is shifting towards this as per their requirements.

- **Mongoose :** Mongoose is a Object Data Modelling(ODM) library for Mongo DB and Node. It manages the relationships between the data , provides schemas validation, and is used to translate between objects in code and the representation of those object in mongo DB. Basically it is the library which help us to store the data in mongo DB but in a efficient way . It internally create all the data given to it to the format in which data is stored in Mongo DB so we donot need to worry about the format of storing data or in other words donot need to format the data in required format manually mongoose handle it smoothly. It provides various function to do manipulation with the database which make our life simple . In this project complete database model is based on this mongoose and the function offered by this is used to do operations . In real world again it is in demand as it give ease to do the things efficiently.

- **Bootstrap:** It is a framework which help us to design websites faster and easier . It includes HTML and CSS based design templates for typography, forms, buttons, tables, navigation, modals, image carousels, etc . It also give support for JavaScript plugin . It also let us do some external changes if we want to in the designing which is offered by it by default. In this project some of the bootstrap is there which help in the design of some of the components there. In real world since it give inbuilt design template to the user it is very much useful. Since developer donot need to do all things with scratch they just need to take the template and modify it as per their requirement and need.

- **Axios :** It is just a JavaScript module or library used to do request to the server like GET, POST, PUT, PATCH , DELETE and will give use the information from the server and will also provide information to the server. In this project all the request are done using this very library and it works in a very efficient manner and easy to use.

- **CDN (Content Delivery Network) :** CDN refers as Content Delivery Network is used to integrate the content from a particular web address to our webpages without actually downloading all the contents explicitly . With the help of CDN we can use the information present in that particular CDN address anywhere if we want and hence it keeps things simple. We donot need to import the complete file or information explicitly, it just provide all those information which we need nothing other than that. In our project we used CDN for axios and able to do request with the help of these CDN link. It is generally used in the real world organization because it keeps thing pretty simple and reduce the wastage of the computer storage.

Reason For Choosing This Technology

This project is one of the important project. There are several reasons to choose this project for the summer term but the most important thing is the interest in the topic. Since one can do nothing if one don't have any interest in the respective topics or subject. It's the interest in the development field either it is web development or it's a android development which act as a driven force for this project . This project teach us about those most basic technologies which one should know if he/she want to be a good developer . A website in a web development contains what?? A webpage whose body is created by HTML and whose designing is done by basically CSS . The functionality of webpage is also very important which is given by JavaScript. Now the most important thing is the information that where we should keep the information and from where we should extract the information , and both the above place refer to the same place called Database. Hence , this very database is build by using Mongo DB in this case. Now the fact is how do we get the information means who will give us information from the database and who will put the information in the database , all these task is performed by something called Server, means server building is also very important. Here Node JS is used to do this server building task. Now the most important after where and who is how we do that operations like keeping data in the database or getting some information from the database who will command the server to do all these. The answer is the endpoints created by using Express in Node JS which we refer to do any such tasks . Hence for overall website all the basic need is fulfilled by using these basic technologies at least for the beginner level. Hence this is chosen Since it give the knowledge of all the technologies/tools mentioned namely HTML , CSS , JavaScript, Mongo DB , Node JS , Express . And this will be the good start if one is willing to be a developer and of course in future he/she can do more with some other good technologies.

CHAPTER -2

Profile of the Problem

This project is basically a “Application management System” Project. The whole project problem is divided into several parts and this project should have all parts inside it. Several Parts are as follows:

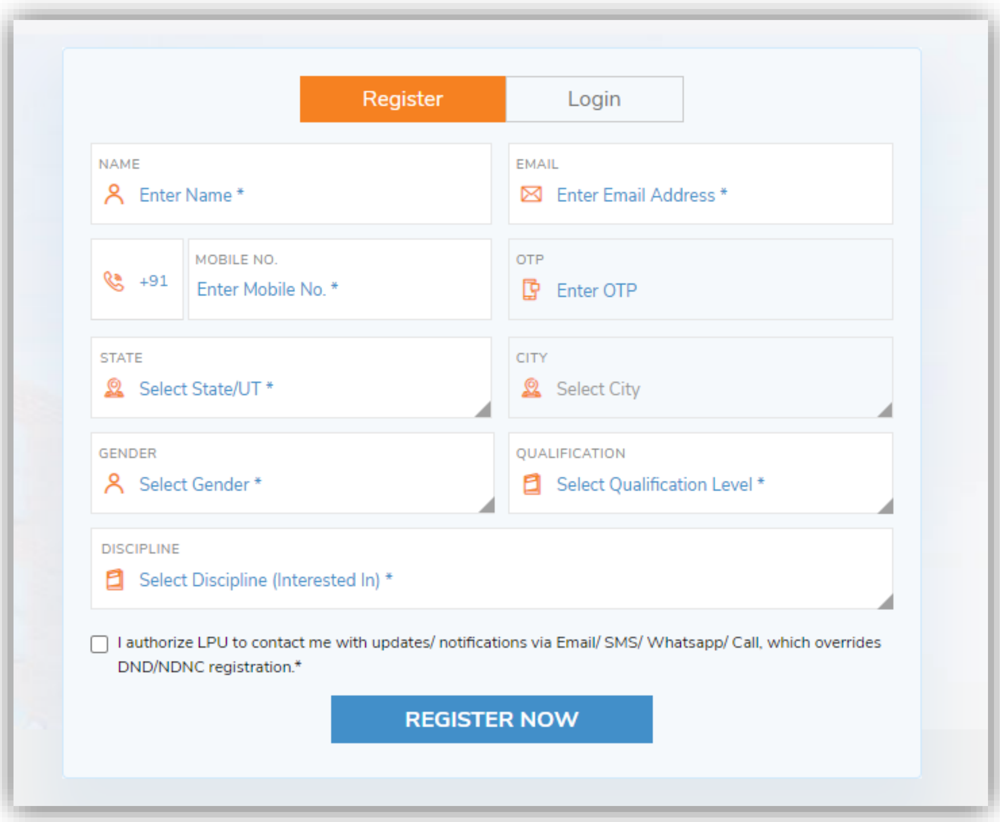
- The first thing is to create a Homepage from where the applicant can Sign Up and Login to visit his/her respective application portal.
- For signup we should take the information like Name, username, email, password to create account and after successful registration the applicant will be assigned a registration number which is unique and one email should have only one account.
- One should be able to login using registration number and password created while registration.
- If someone forgot his/her registration number or password he/she will able to forgot those from the forgot link provided by entering their respective email id in the home page.
- After successful registration user should be able to fill the form which is asking several entries from the user like Name, Father's Name, Mother’s Name, Date of Birth, Gender, Identity Type, Identity Number, Photograph, Address, Locality, City/town/village, District , State , Country , Pin Code , Email and phone number . And after confirming all the details the details will save to the database.
- After successful registration the application preview page should open where user can see all the details he/she entered and if wanted can also be able to edit all the details using edit details button. And after the update of details again the whole updated data is updated in the database.
- Now one can also print the screen of the preview window as per the acknowledgement page for further uses.
- One can also be able to change the password if wanted after login , from the change password button which is there in the application, preview and edit details window .
- One can also be able to logout using logout button and redirected to the home window.
- One special condition is that if user get registered and filled the application then again after login he/she should be directed to preview window directly but if he/she is not filled the form , must be directed to application window first . Where he/she can perform the respective operation as per their choice mention above.
- The above is the whole statement of the problem.

Existing system

This project is based on “Application Management System” which is based on NodesJs, Mongo DB, Express etc. Talking about existing system there are many real world examples we can take where the whole logic which is implemented in this project are actually present in the current world .As in this very modern world no any work can be done without human power and for this selection of the human resource is also very important and for all this in the whole world different types of application system are present where people like us come and enroll and try to grab the opportunities they are offered. As it is very clear that whole world is dependent on this type of application system for some reasons. There are many examples we can take to understand or explain the existing system related to that . Let us take the example of the admission process in any college or universities.

In any college and universities , the whole process is based in this application management system. While admissions colleges used to take online exams of the candidate where they use this technology to register student for the examination by taking their details like name , father’s name, mother’s name , email, phone number, address , photo etc. and provide them a unique application number or ID for the examination . Again after examination while counseling they used to take details and concern from the student to register them as their student for some course and this just goes on while the admission get successful. This whole process use a very crucial technique or application of a “ student application management system”. Hence it is also very useful application in the modern day.

There are many more existing system of this like Entrance examinations, General Competitive Examinations and many more which is impossible to explain each and every system , Many big giants companies used to use this system in their companies for the convenience in their recruitment process. Hence in general the above is some points which describe the existing system related to this project .



□ I authorize LPU to contact me with updates/ notifications via Email/ SMS/ Whatsapp/ Call, which overrides DND/NDNC registration.*

Fig 2.01 Basic Application Portal

CHAPTER 3

Problem Analysis

The whole problem is basically based on the efficient implementation of database which in this case in Mongo DB using Mongoose. Since in the whole problem we are required to play with information like to get the information correctly and then to store it in such a way that we should be able to extract the information whenever we want to use that. We should also take care about the plagiarism may be possible when registering the user and most importantly we have to give them the unique application number for the further process.

❖ Product Definition

The whole product is basically a combination of webpages integrated with backend where the information feed by the user can be collected and saved in the database via following processes:

- ➢ Signup by Providing Necessary Details and get Unique Application Number.
- ➢ Can sign in using Application Number and Password
- ➢ Can able to forgot the details whenever needed
- ➢ Can fill the registration form.
- ➢ Can able to update the details in the registration form
- ➢ Can be able to change the password when needed
- ➢ Can able to take the print of the application screen
- ➢ Can able to sign out off the interface.

❖ Feasibility Analysis

- ➢ The whole project seems simple but to implement this we do need good understanding of the context and technologies which should be used in the fulfillment of this project. This project need some online platform to store the data so we do need to take care of that also as it is using Mongo DB Atlas to store the data so we do need to take care of that also.

- ➢ If this project need to be hosted then we should take care of getting a particular domain to host it there. Since there is very less platform which provide free domain to use and some who are giving free domain the memory is limited for the free trail and we should also find a platform to store the photos as very less platform allow it.

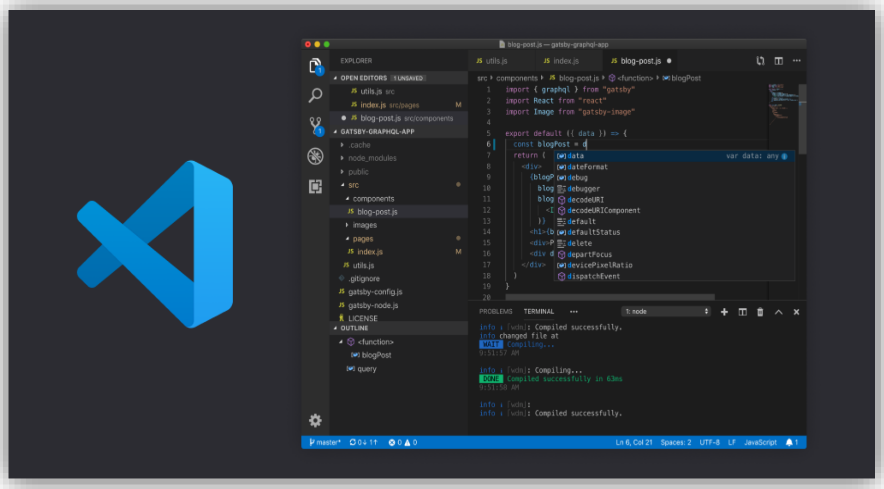
Software Requirements Analysis

Software requirements analysis is one of the major things we should take care before starting this project. Since only learning the technologies is not enough. To implement that we should have some good software where we can actually able to implement the logics and create the required project as per our wish and requirements. Since there are number of software available in the market or internet so we should sincerely choose those which is more efficient and helpful for the one who is trying to develop the project.

As per above brief description this project also holds some software with it which are used to develop the whole project which are mentioned below:-

- Visual Studio Code
- Node JS
- Mongo DB
- Browser
- POSTMAN Postman

- **Visual Studio Code** : The software used for coding the program is the Visual Studio Code Editor. It is a code editor made by Microsoft for Windows, Linux and mac OS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. Visual Studio code helps us in different ways to write the code efficiently and also have integrated terminal to run the code there which is also very useful . So we don’t need to move here and there to run the code all the configurations are inbuilt in the Visual Code Editor , just we have to install the required extensions.

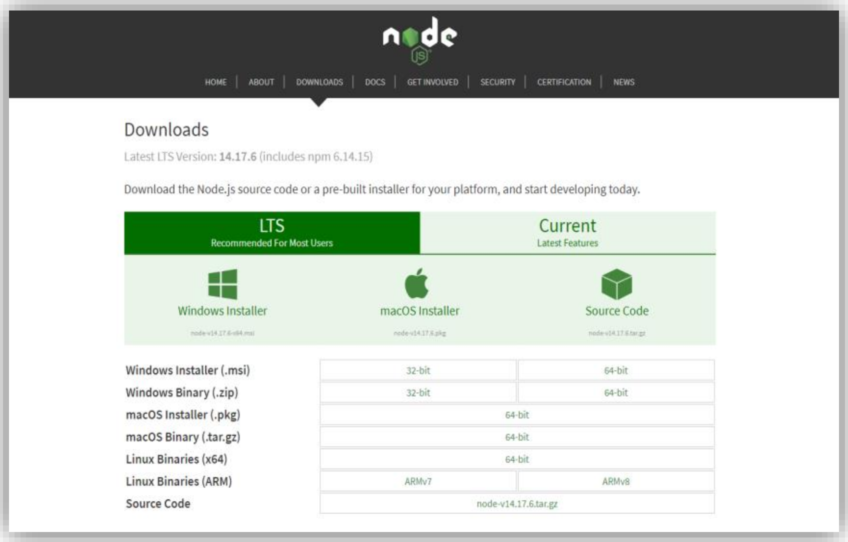


```
import { graphql } from "gatsby"
import React from "react"
import Image from "gatsby-image"

export default ({ data }) => {
  const blogPost = d
  return (
    <div>
      {blogP
        blog
        blog
          <I
        ))
      <h1>{b
      <div>P
      <div d
    </div>
  )
}
```

```
info i [wdm]: Compiled successfully.
info changed file at
WAIT Compiling...
9:51:57 AM

info i [wdm]: Compiling...
DONE Compiled successfully in 63ms
9:51:58 AM

info i [wdm]:
info i [wdm]: Compiled successfully.
```

Fig 3.01 Visual Studio Code Interface

❖ **Node JS** : This is another software this project is using for using the Node JS logic and NPM Packages.Node.js (Node) is an open source development platform for executing JavaScript code server-side. Node is useful for developing applications that require a persistent connection from the browser to the server and is often used for real-time applications such as chat, news feeds and web push notifications. In this project Node JS is used to make the backend Server where we can do all the backend logic and also for the very important application of the express for making calls to the endpoint in the backend sever.



Downloads

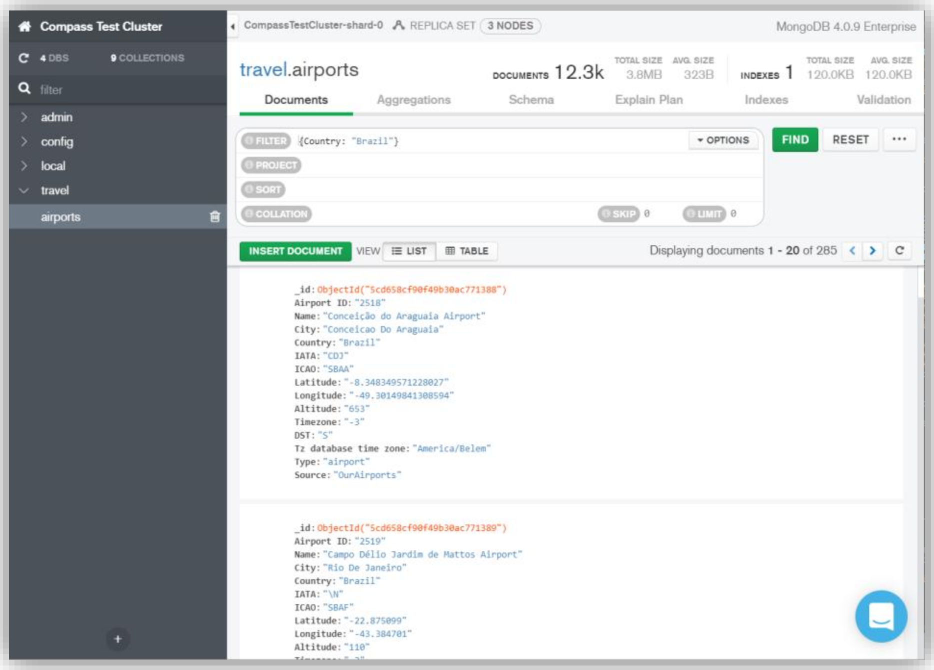
Latest LTS Version: 14.17.6 (includes npm 6.14.15)

Download the Node.js source code or a pre-built installer for your platform, and start developing today.

Windows Installer (.msi)	32-bit	64-bit
Windows Binary (.zip)	32-bit	64-bit
macOS Installer (.pkg)	64-bit	
macOS Binary (.tar.gz)	64-bit	
Linux Binaries (x64)	64-bit	
Linux Binaries (ARM)	ARMv7	ARMv8
Source Code	node-v14.17.6.tar.gz	

Fig 3.02 Node JS Interface

❖ **Mongo DB :** This is a another software this project is using ,for reviewing the data in the database where we are storing the information. Mongo DB Compass is a powerful GUI for querying, aggregating, and analyzing your Mongo DB data in a visual environment. Compass is free to use and source available, and can be run on mac OS, Windows, and Linux. This is very helpful as it provide the direct access to all the information locally so that we can able to verify whether all the operations are working fine or not.



travel.airports

DOCUMENTS 12.3k TOTAL SIZE 3.8MB AVG. SIZE 323B INDEXES 1 TOTAL SIZE 120.0KB AVG. SIZE 120.0KB

FILTER {Country: "Brazil"}

SKIP 0 LIMIT 0

Displaying documents 1 - 20 of 285

```
_id: ObjectId("5cd658cf90f49b30ac771388")
Airport ID: "2518"
Name: "Conceição do Araguaia Airport"
City: "Conceicao Do Araguaia"
Country: "Brazil"
IATA: "CDJ"
ICAO: "SBAA"
Latitude: "-8.348349571228027"
Longitude: "-49.30149841308594"
Altitude: "653"
Timezone: "-3"
DST: "S"
Tz database time zone: "America/Belem"
Type: "airport"
Source: "OurAirports"

_id: ObjectId("5cd658cf90f49b30ac771389")
Airport ID: "2519"
Name: "Campo Délio Jardim de Mattos Airport"
City: "Rio De Janeiro"
Country: "Brazil"
IATA: "\N"
ICAO: "SBAF"
Latitude: "-22.875099"
Longitude: "-43.384701"
Altitude: "110"
```

Fig 3.03 Mongo DB Compass Interface

❖ **Browser :** Browser is one of the essential component we are using in this project. A web browser, or simply 'browser,' is an application used to access and view websites. Common web browsers include Microsoft Edge, Internet Explorer, Google Chrome, Mozilla Firefox, and Apple Safari. The primary function of a web browser is to render HTML, the code used to design or 'mark up' webpages". Browser are used to see the User Interface (UI) which we are making and it also act as a platform for the execution of our website. For this project we can use any browser like Google Chrome , Mozilla Firefox or any other. But we prefer to use Google Chrome as it supports most of the JavaScript logic that we used in this project as it has V8 JavaScript Engine.

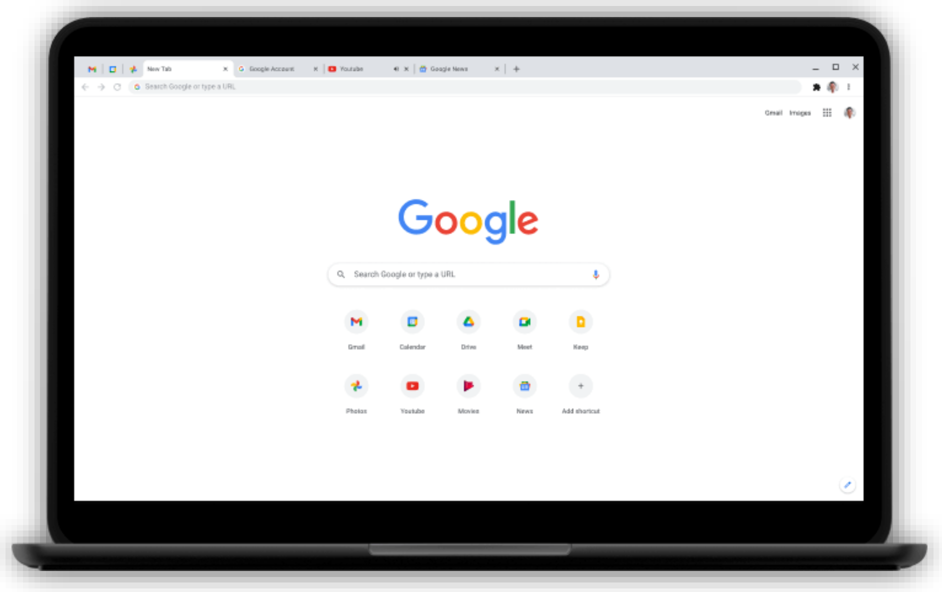
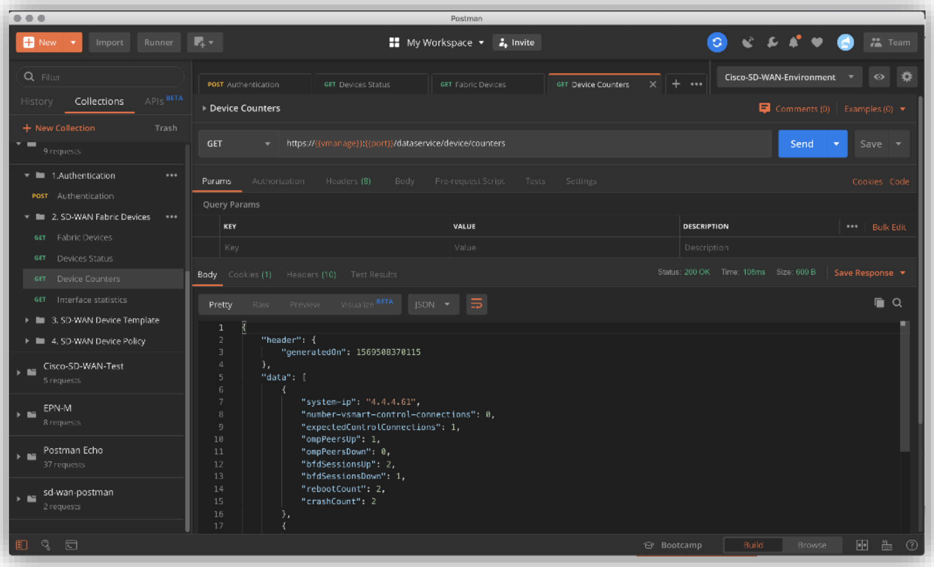


Fig 3.04 Google Chrome Browser

- POSTMAN **Postman :** This is one of that software which is in very much demand among all the developers. Postman is an API client that makes it easy for developers to create, share, test and document APIs. This is done by allowing users to create and save simple and complex HTTP/s requests, as well as read their responses. The result - more efficient and less tedious work This software help us to do all the REST API requests like GET, POST, PATCH, PUT, DELETE to the specific end point without having a front end. This help us to do all those things directly and we can do all things such that passing data of any format as we want and can able to see the result coming from the endpoint.

This help us to check and validate all the request and response before doing it in the actual code with User Interface (UI).



https://{{vmanage}}:{{port}}/dataservice/device/counters

Status: 200 OK Time: 108ms Size: 609 B

```
{
    "header": {
        "generatedOn": 1569508370115
    },
    "data": [
        {
            "system-ip": "4.4.4.61",
            "number-vsmart-control-connections": 0,
            "expectedControlConnections": 1,
            "ompPeersUp": 1,
            "ompPeersDown": 0,
            "bfdSessionsUp": 2,
            "bfdSessionsDown": 1,
            "rebootCount": 2,
            "crashCount": 2
        },
        {
```

Fig 3.05 Postman Software Interface

- All the software mention above is very important for our whole project as these provides the necessary development environment for our project. It give us the ease that we can able to concentrate on the logical part of the project more instead of focusing on the syntactical part . Moreover these software are also very helpful for the debugging process . It provides many ways to debug our code and gives us the best possible suggestion for debugging.

It is not necessary that we should only use these combination of the software , we can use various other text editors like Sublime, Brackets and many more as per our interest, we can also use other databases like MySQL , SQLite , Graph QL etc. for storing the data but in this case we use Mongo DB as per our interest. This is whole about software used , now let we move in further part of the report where the description of the whole project is provided.

CHAPTER 4

Design Of The Project

Design is one of the major and very important factor that should be taken care of before starting any type of development. This project also need to be designed for the proper implementation and programming of the entire project. The whole project is divided into several sections for front end and for the backend separately as mentioned below:

- **For Frontend (Client Side)**

  The below mentioned section are designed for the frontend UI

  - **Home Page**: Home page is the first webpage of the project. In the home page at first there is a Heading after that there are some buttons and links available which we can use to navigate or to open some other view in the navigation bar . In the homepage we can able to login , Signup and forgot the password if needed because of the separate window available for the same there. The signup and the sign in and forgot window have some implementation which are discussed further.

  - **Sign In window**: In login window user need to enter registration or application number provided to them and a unique password to login.

  - **Sign Up Window :** In signup window user need to register by providing the required information asked there and then after submitting the details they get the newly and uniquely generated application number and do the further process

  - **Forgot Password Window :** Here in the forgot password window if user somehow not able to remember his/ her password or registration number then he/she can just enter the unique email there and can able to retrieve the application number and the password from the database.

- **Application Window:** In the application window there is a bar where the user application number will be reflected and then there is a edit password and logout button which user can use as per his/her wish. Further there is a form available which the user should fill to get registered completely which is divided in to some section like Personal Details, Contact Information, Confirmation which is a part of UI. There are button for the user to submit by which they can submit the application.

- **Update Details Window :** This window is just same as application window the only difference is here the information is auto filled by the data provided by the user earlier but here user can actually able to edit and update the details which so ever they want. By submitting the button the data will be updated.

- **Preview Window:** It is a window where user can able to see the preview of the details which is submitted by them while filling the application and if they want to edit or update the details they can use the update details button provided there to update the details else they can take the print out of the screen by the print button provided there.

- **Change Password Window :** The change password window is very helpful for the user , it is available on the application, update application and preview window and whenever user want to change the password user can make use of the change password button there and can actually change the password by entering some of the details asked there.

- **Logout Button :** Logout button is one of the essential part. By clicking the logout button user can able to logged out of the portal and re-directed to the homepage. Once the user click the logout button a dialog box will appear and by confirming it there the get logged out of the portal.

- ❖ **For Backend (Server Side)**

  Backend is also the major part of the project and it is also as important as the frontend .At first we design the server using Node JS and then after there are 2 aspects remains of designing of the backend as mentioned below:

  - ✓ *Collections for Storing the data in the Database.*

  - ✓ *End points through which we can actually able to save information to the database and retrieve information from the database by calling the required endpoint through necessary REST API request to the endpoint.*

  - **Designing of the collection :** Collection is the most important part of the Mongo DB Database , it define the structure how the data is going to store in the database. There are 2 types of collections here made in the backend first one is with the name of *"applications"* which store the application information provided by the user in the collection and the second collection is with the name of *"registrations"* which store the registration details means only the Signup related information about the user.

  - **Designing of the endpoints :** End points is one of the major part of backend again as it provide the information required whenever call wisely with a correct REST API request. There are several endpoints created in the sever for doing CRUD (create, read, update, delete) with the database collections whenever needed in the whole application. The only thing we have to do is to call the endpoints through fetch API or through Axios from the frontend and it will do the required job for us and after that we can able to manipulate the data as per needed.

Now after the designing process we can move the the most important and exciting part that is implementation process in the next section .

Flow Chat For the Front-End (Client Side)

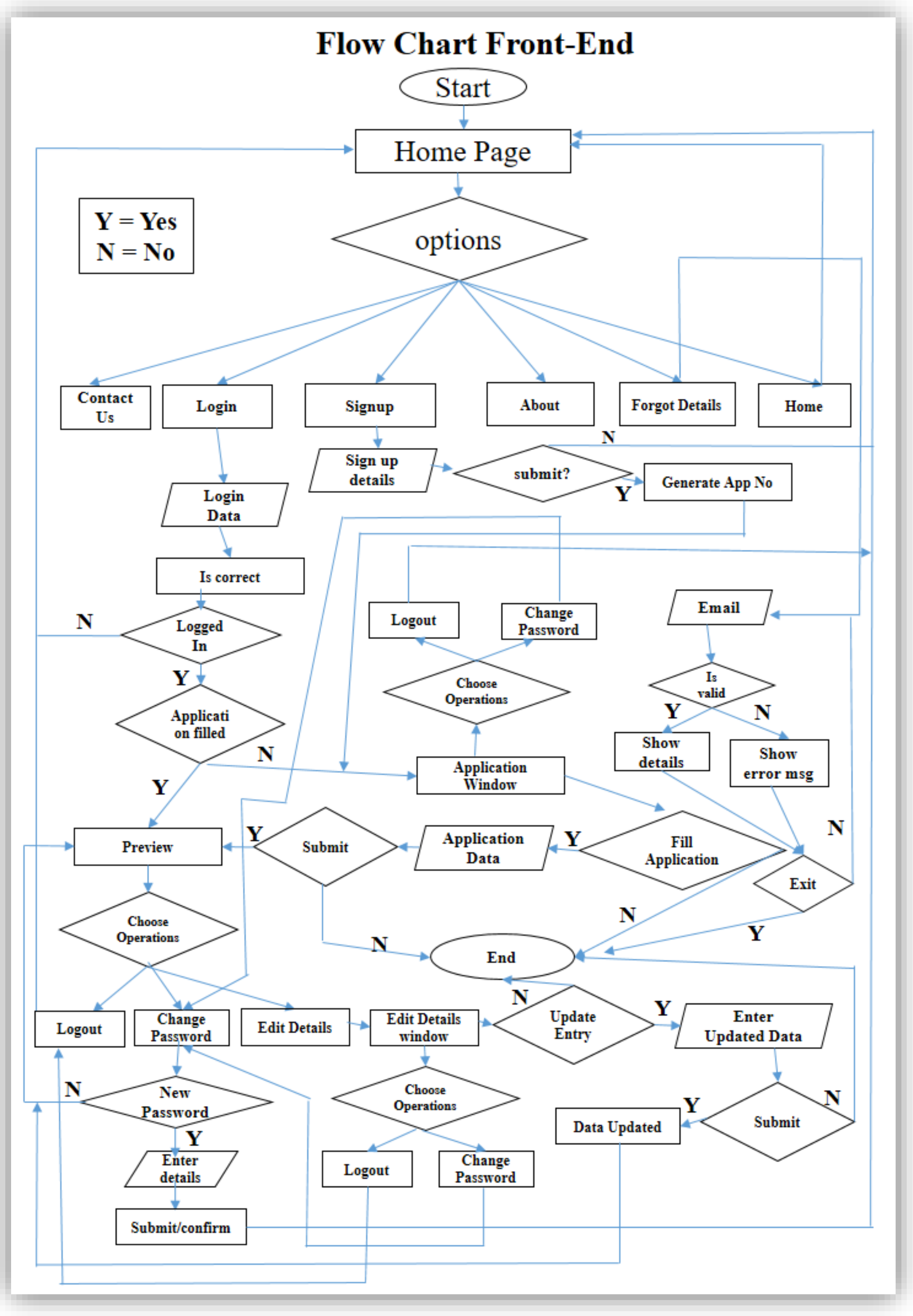


Fig 4.01 Client Side Flow Chart(Front End)

Flow Chat For the Back-End (Server Side)

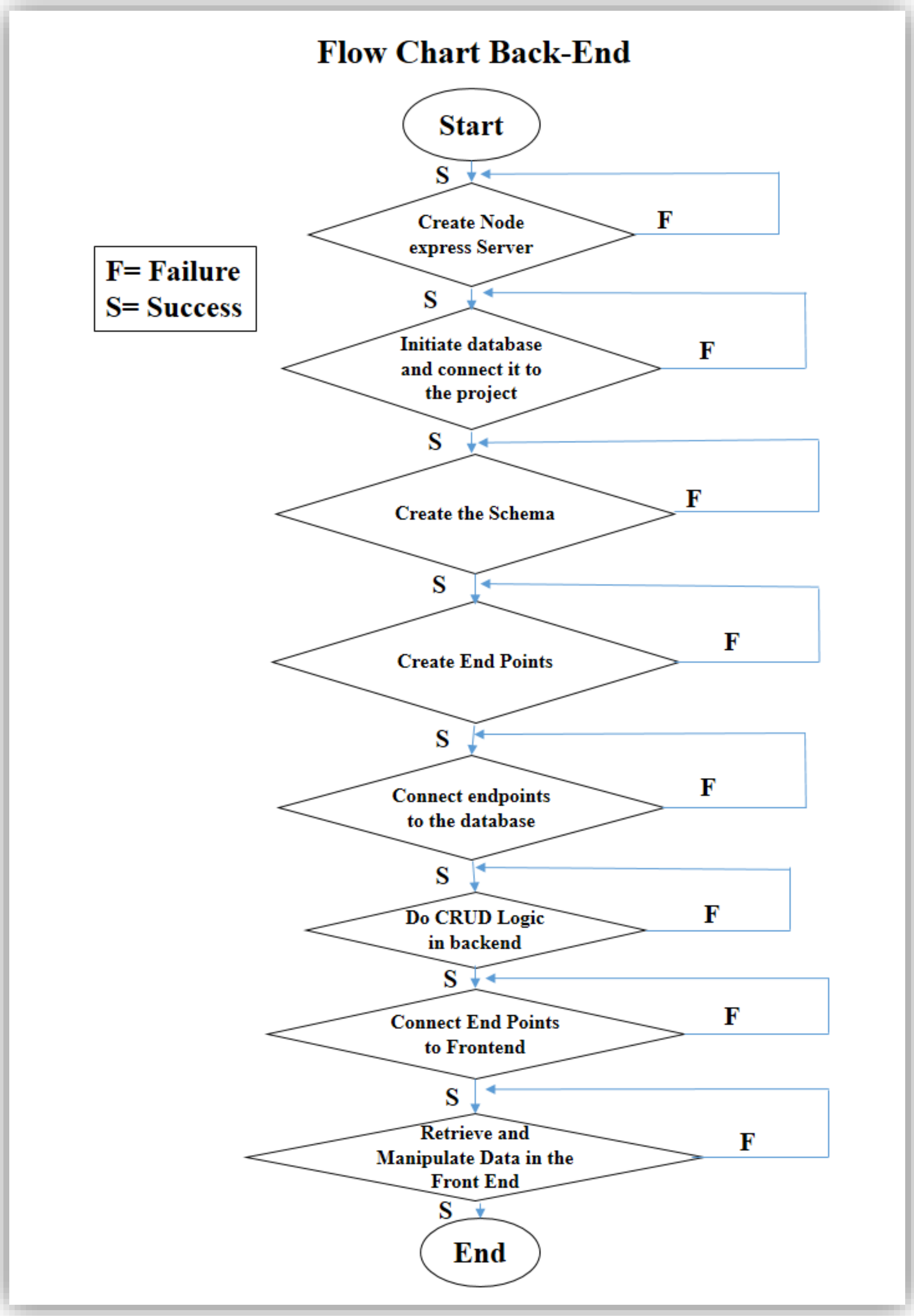

Flow Chart Back-End
Start
F= Failure
S= Success
S
Create Node express Server
F
S
Initiate database and connect it to the project
F
S
Create the Schema
F
S
Create End Points
F
S
Connect endpoints to the database
F
S
Do CRUD Logic in backend
F
S
Connect End Points to Frontend
F
S
Retrieve and Manipulate Data in the Front End
F
S
End

Fig 4.02 Server Side Flow Chart(Back End)

IMPLEMENTATION

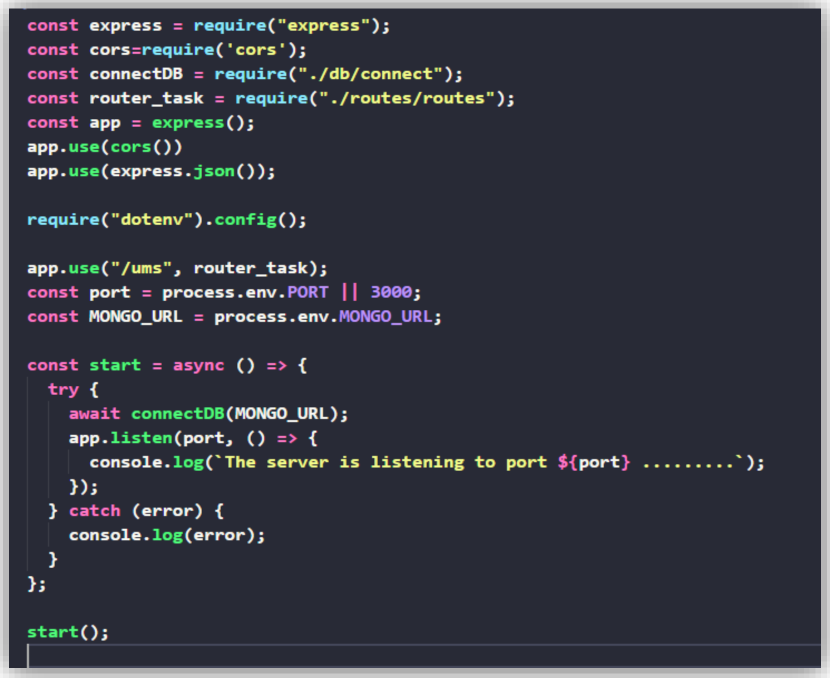
This section is completely about the implementation , means how we actually going to implement all the logics , design , structures mentioned above in the design section in the actual code. Here we are going to discuss each and every aspect of implementation step by step so that we can have a clarity how all the process goes. For this, from now onwards the whole implementation section is subdivided into several subsections for the better understanding and the clarity.

Now, from here the implementation of both the Front End and the Back End is described in separate Section and at Last the working flow of the project has discussed. So, at first we will going to discuss about the Implementation of the Back End Since it is necessary for the implementation of the Front End and not vice-versa.

Backend Implementation

The whole backend implementation is divided into several parts as follows:

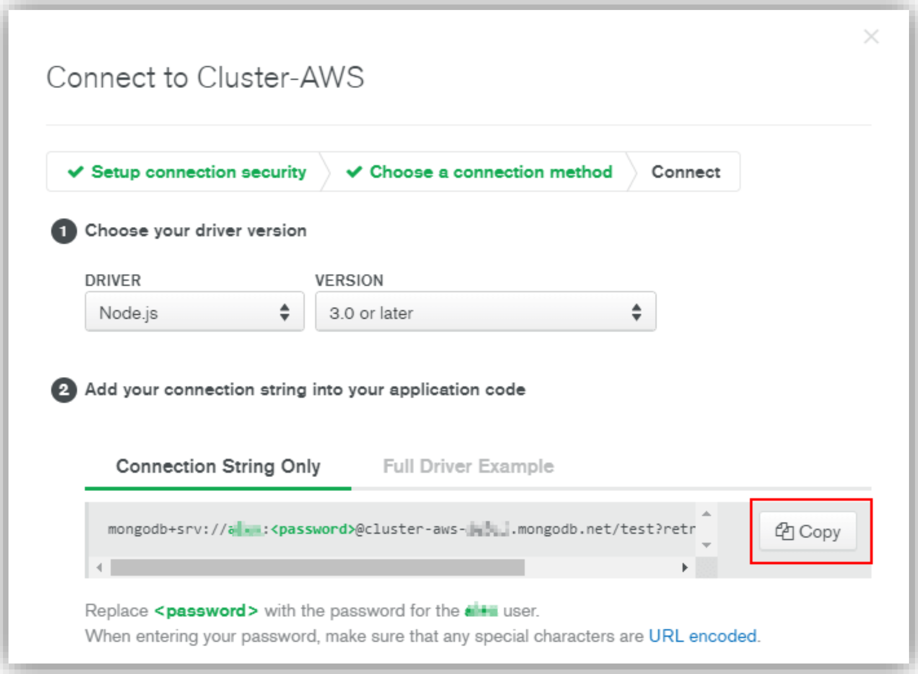
- **Create Node JS Server :** At first we create the express App for our project and Node JS express Server by using express Module which inbuilt create a Server and by this we can able to use the server for listening the different request at a specific port number. The basic code of the creating the express app is given below.



```javascript
const express = require("express");
const cors=require('cors');
const connectDB = require("./db/connect");
const router_task = require("./routes/routes");
const app = express();
app.use(cors())
app.use(express.json());

require("dotenv").config();

app.use("/ums", router_task);
const port = process.env.PORT || 3000;
const MONGO_URL = process.env.MONGO_URL;

const start = async () => {
  try {
    await connectDB(MONGO_URL);
    app.listen(port, () => {
      console.log(`The server is listening to port ${port} .........`);
    });
  } catch (error) {
    console.log(error);
  }
};

start();
```

Fig 4.03 Express Code for creating app

- **Connect with the Database :** In this step we have to connect with the database , we can use any kind of database like MySQL or Mongo DB as per out choice but for this project we are using the Mongo DB database. Here Mongo DB atlas is used to create the database and collections. It will give the connection string that will be use to connect it to project. Mongoose is used to do all things with the database which comes with the Mongo DB Client. It make our task very easy by doing complex and detailed operations implicitly internally. So we don't need to write the detailed code for that .A basic view for the setup of Cluster and Getting Connection String is given below.



Connect to Cluster-AWS

✓ Setup connection security ✓ Choose a connection method Connect

1 Choose your driver version

DRIVER: Node.js

VERSION: 3.0 or later

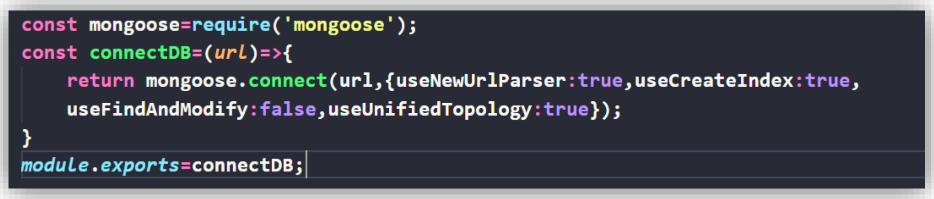
2 Add your connection string into your application code

```
mongodb+srv://a[redacted]:<password>@cluster-aws-[redacted].mongodb.net/test?retr
```

Replace **<password>** with the password for the [redacted] user.
When entering your password, make sure that any special characters are URL encoded.

Fig 4.04 Mongo DB Atlas Snip

- **Connect Database to the Project** : In this part the database created or the cluster for the database created earlier in the Mongo DB Atlas is now connected to the project with the help of the Mongoose and the Connection String.



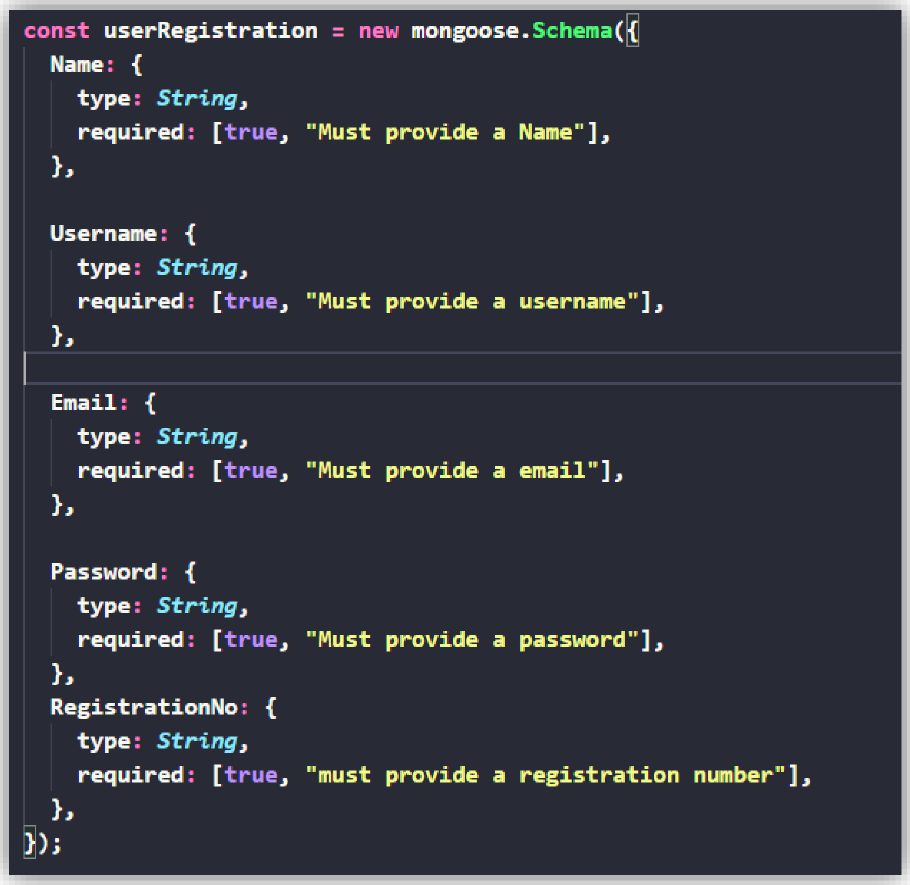
```
const mongoose=require('mongoose');
const connectDB=(url)=>{
    return mongoose.connect(url,{useNewUrlParser:true,useCreateIndex:true,
    useFindAndModify:false,useUnifiedTopology:true});
}
module.exports=connectDB;
```

Fig 4.05 Mongo DB Atlas Snip

- **Creating Schemas :** Now here the create of the Schemas of the collection happened. There are 2 Schemas in this project first for registrations and second for the applications as detailed description is given below:

✓ **For User Registration**
In this schema we are collection or saving the registration related details of the user and also allocating a unique registration number using “uuid NPM Module”. The entry of the schemas mentioned below:

  - Name
  - User Name
  - Email
  - Password
  - Registration Number



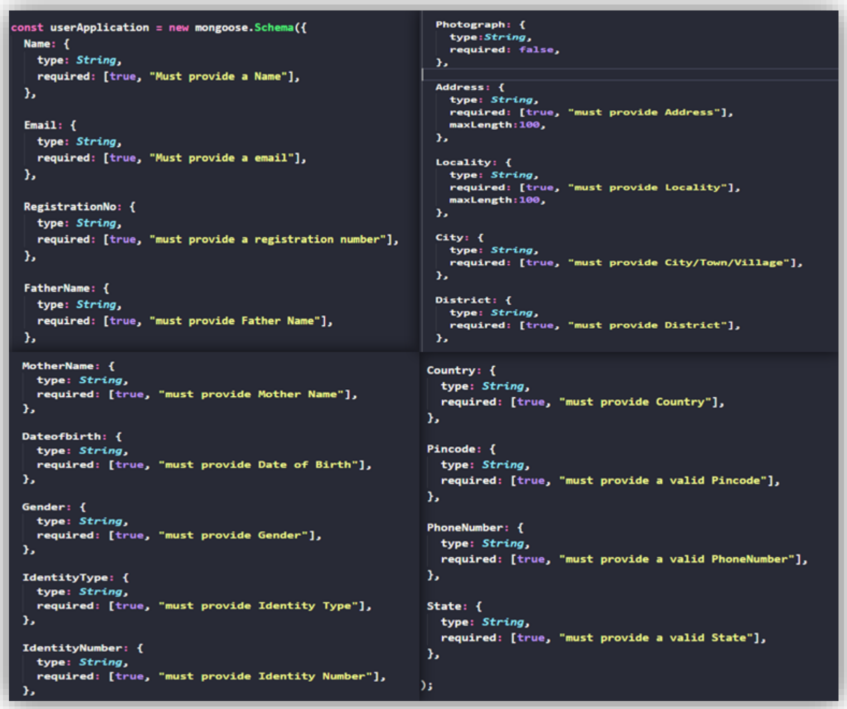
```
const userRegistration = new mongoose.Schema({
  Name: {
    type: String,
    required: [true, "Must provide a Name"],
  },

  Username: {
    type: String,
    required: [true, "Must provide a username"],
  },

  Email: {
    type: String,
    required: [true, "Must provide a email"],
  },

  Password: {
    type: String,
    required: [true, "Must provide a password"],
  },
  RegistrationNo: {
    type: String,
    required: [true, "must provide a registration number"],
  },
});
```

Fig 4.06 User Registration Schemas

✓ **User Application**

This is the schema for the user application details where we are actually storing the Application form details if entered by the user for the successful submission of the application. The entries related to this schema is mentioned below:

- Name
- Email
- Registration Number
- Father’s Name
- Mother’s Name
- Date Of Birth
- Gender
- Identity Type
- Identity Number
- Photograph
- Address
- Locality
- City
- District
- Country
- Pin code
- Phone Number
- State



```
const userApplication = new mongoose.Schema({
  Name: {
    type: String,
    required: [true, "Must provide a Name"],
  },

  Email: {
    type: String,
    required: [true, "Must provide a email"],
  },

  RegistrationNo: {
    type: String,
    required: [true, "must provide a registration number"],
  },

  FatherName: {
    type: String,
    required: [true, "must provide Father Name"],
  },

  MotherName: {
    type: String,
    required: [true, "must provide Mother Name"],
  },

  Dateofbirth: {
    type: String,
    required: [true, "must provide Date of Birth"],
  },

  Gender: {
    type: String,
    required: [true, "must provide Gender"],
  },

  IdentityType: {
    type: String,
    required: [true, "must provide Identity Type"],
  },

  IdentityNumber: {
    type: String,
    required: [true, "must provide Identity Number"],
  },

  Photograph: {
    type:String,
    required: false,
  },

  Address: {
    type: String,
    required: [true, "must provide Address"],
    maxLength:100,
  },

  Locality: {
    type: String,
    required: [true, "must provide Locality"],
    maxLength:100,
  },

  City: {
    type: String,
    required: [true, "must provide City/Town/Village"],
  },

  District: {
    type: String,
    required: [true, "must provide District"],
  },

  Country: {
    type: String,
    required: [true, "must provide Country"],
  },

  Pincode: {
    type: String,
    required: [true, "must provide a valid Pincode"],
  },

  PhoneNumber: {
    type: String,
    required: [true, "must provide a valid PhoneNumber"],
  },

  State: {
    type: String,
    required: [true, "must provide a valid State"],
  },

});
```

Fig 4.07 User Application Schemas

- **Creating Endpoints Or Routes** : This section is related to the creation of the endpoints or we can say routes for listening the request pass by the user by the server. These routes are helpful in doing backend CRUD Operations to and fro from the backend to frontend and vice-versa. These routes are linked with the server so that any request made with specific route will result the desired result for the client. There are basically 4 major operation can be done with the data that is Creating Data, Reading the Data, Updating The Data and Deleting the Data (CRUD). This project is using 3 of these operations and having specific endpoint for different CRUD operations as mentioned below:

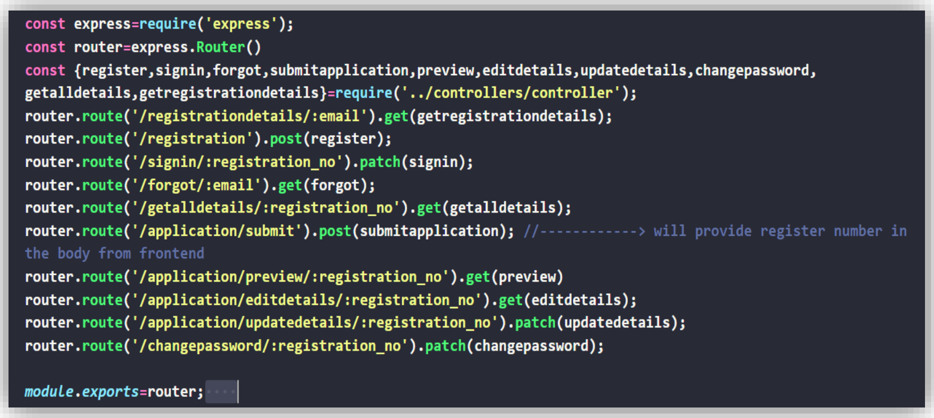
  In backend there operations are done with the help of some specific REST API Request as mentioned:

  - ✓ Creating The Data In Database : POST Request
  - ✓ Reading The Data From Database : GET Request
  - ✓ Updating The Data In The Database : PUT Or PATCH Request.
  - ✓ Deleting The Data From The Database : DELETE Request.

- ➢ **Endpoints related to the Creating Data (POST Request)**
  The endpoints or the routes which help us to put specific info in the database are mentioned below:
  - **/ums/registration** – for registration
  - **/ums/application/submit** –for submitting applications

- ➢ **Endpoints related to Reading the Data (GET Request)**
  - **/ums/registrationdetails/:email** -for getting reg.details of a user with given unique email.
  - **/ums/forgot/:email** – for forgetting details of a user with specific unique email
  - **/ums/getalldetails/:registration_no** – for getting all detail of the user with specific unique registration
  - **/ums/application/preview/:registration_no** – for preview of the details of the user with specific registration number.
  - **/ums/application/editdetails/:registration_no**-for putting data in the preview for the user with specific registration number.

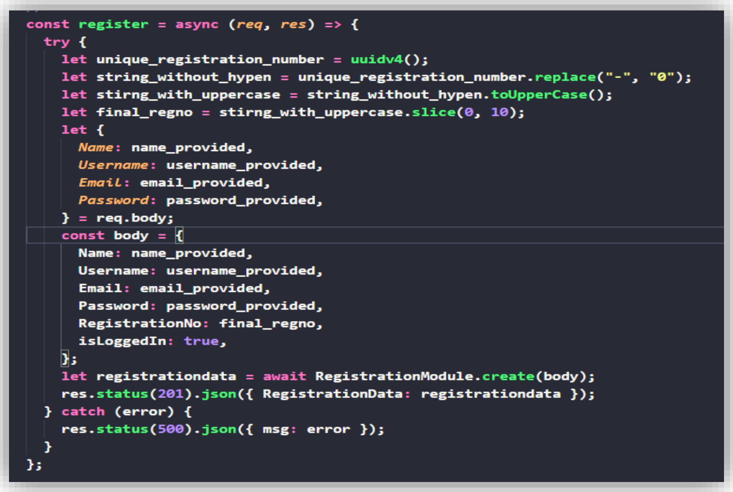
- **Endpoints related to Updating The Data(PATCH Request)**
  - **/ums/signin/:registration_no** – for updating singin details of the user with specific registration number
  - **/ums/application/updatedetails/:registration_no** –for updating application details for the user with specific registration number
  - **/ums/changepassword/:registration_no** – for changing password for the user with specific registration.



```
const express=require('express');
const router=express.Router()
const {register,signin,forgot,submitapplication,preview,editdetails,updatedetails,changepassword,
getalldetails,getregistrationdetails}=require('../controllers/controller');
router.route('/registrationdetails/:email').get(getregistrationdetails);
router.route('/registration').post(register);
router.route('/signin/:registration_no').patch(signin);
router.route('/forgot/:email').get(forgot);
router.route('/getalldetails/:registration_no').get(getalldetails);
router.route('/application/submit').post(submitapplication); //------------> will provide register number in
the body from frontend
router.route('/application/preview/:registration_no').get(preview)
router.route('/application/editdetails/:registration_no').get(editdetails);
router.route('/application/updatedetails/:registration_no').patch(updatedetails);
router.route('/changepassword/:registration_no').patch(changepassword);

module.exports=router;
```

Fig 4.08 Endpoints/routes

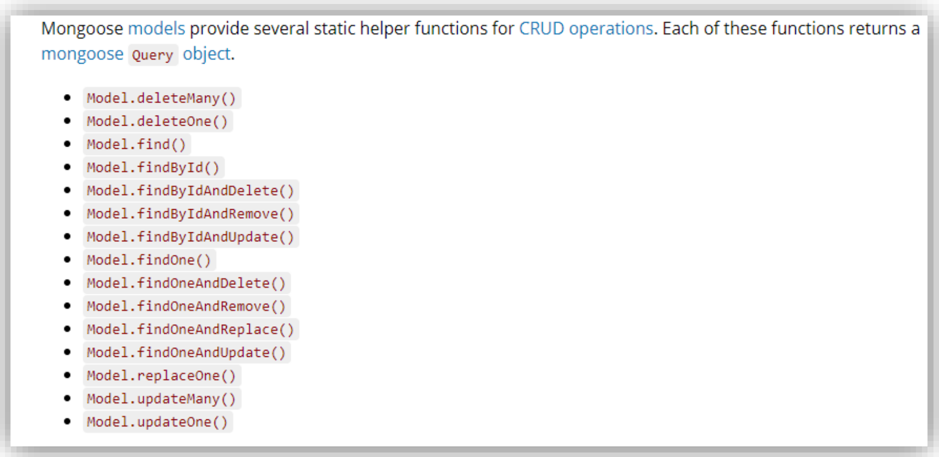
- **Connecting End Points with The Database:** In this step we have to connect these endpoints with the database with the help of controllers where we actually able to do CRUD operations with the database. For this purpose we are Using Mongoose client provided by the Mongo DB database which make our work easier. A sample snippet of the process is given below for more reference visit the link provided in the end of implementation section to see the actual code.



```
const register = async (req, res) => {
  try {
    let unique_registration_number = uuidv4();
    let string_without_hypen = unique_registration_number.replace("-", "0");
    let stirng_with_uppercase = string_without_hypen.toUpperCase();
    let final_regno = stirng_with_uppercase.slice(0, 10);
    let {
      Name: name_provided,
      Username: username_provided,
      Email: email_provided,
      Password: password_provided,
    } = req.body;
    const body = {
      Name: name_provided,
      Username: username_provided,
      Email: email_provided,
      Password: password_provided,
      RegistrationNo: final_regno,
      isLoggedIn: true,
    };
    let registrationdata = await RegistrationModule.create(body);
    res.status(201).json({ RegistrationData: registrationdata });
  } catch (error) {
    res.status(500).json({ msg: error });
  }
};
```

Fig 4.09 Snip of the controller connecting endpoint to database

- **CRUD Operations form the Backend :** In this section we actually do the CRUD operations for different endpoints as per required. These whole operation can be done with the help of Mongoose Client provided by the Mongo DB database Software. The basic functions which are used to do CRUD Operations are mentioned below:



Mongoose models provide several static helper functions for CRUD operations. Each of these functions returns a mongoose `Query` object.

- `Model.deleteMany()`
- `Model.deleteOne()`
- `Model.find()`
- `Model.findById()`
- `Model.findByIdAndDelete()`
- `Model.findByIdAndRemove()`
- `Model.findByIdAndUpdate()`
- `Model.findOne()`
- `Model.findOneAndDelete()`
- `Model.findOneAndRemove()`
- `Model.findOneAndReplace()`
- `Model.findOneAndUpdate()`
- `Model.replaceOne()`
- `Model.updateMany()`
- `Model.updateOne()`

Fig 4.10 Mongoose CRUD Functions

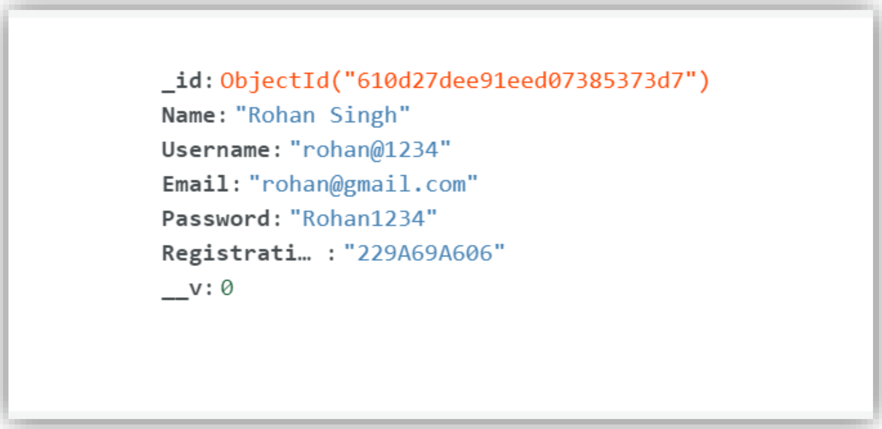
- **Connect End Points To the Front End :** In this section we discuss the connection of the endpoints to the front End. It is one of the major and important steps to do because without doing this step our whole efforts just get wasted. For this connection we can use the “Fetch API” or “Axios” in the front End Javscript where we can actually call and pass the required body to the endpoints and do the desired operations.

- **Retrieve and Manipulate Data From the Front End :** This is the last step in the backend development. This step is about how we actually retrieve information from the server and send information to the server. By using all the above steps now we are ready with our backend and now we can start our server and can do whatever operations we want to do from the front End.

- Now after the backend development we can put it on the one side and can move forward to go with the development of the frontend and after the development of the frontend we can able to run both the front end and the backend with the specific servers. **Note:--IN ALL THE BACKEND WE ARE DEALING WITH THE JSON DATA MEANS WE STORE JSON DATA, GIVE JSON DATA AND FETCH JSON DATA.**

- Now after the above discussion with the backend. It is important that we should get a glimpse of how actually the data looks inside the database. So below given some of the snaps of the data from the collection to get an idea of what things really seems to be.

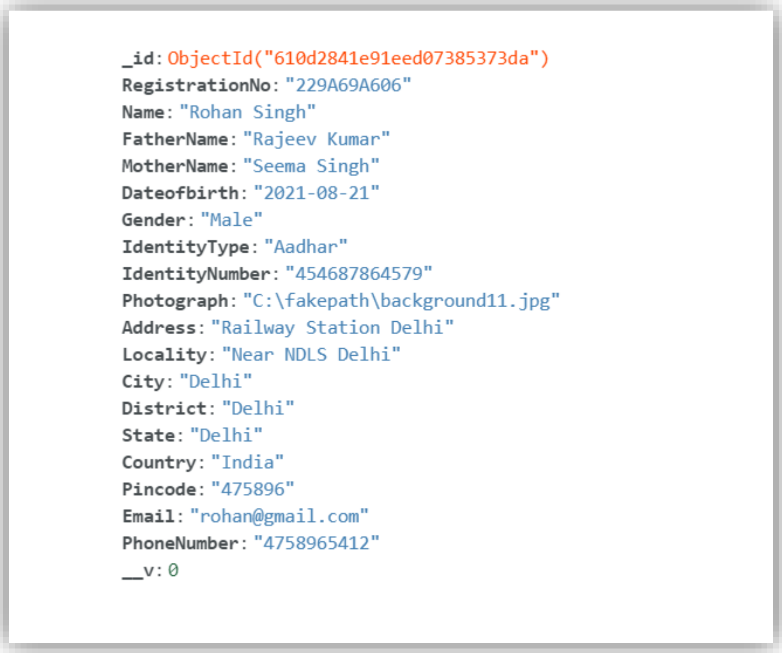
  - **Entry for registration**



```
_id: ObjectId("610d27dee91eed07385373d7")
Name: "Rohan Singh"
Username: "rohan@1234"
Email: "rohan@gmail.com"
Password: "Rohan1234"
Registrati… : "229A69A606"
__v: 0
```

Fig 4.11 Registration data entry

  - **Entry for Application Submission**



```
_id: ObjectId("610d2841e91eed07385373da")
RegistrationNo: "229A69A606"
Name: "Rohan Singh"
FatherName: "Rajeev Kumar"
MotherName: "Seema Singh"
Dateofbirth: "2021-08-21"
Gender: "Male"
IdentityType: "Aadhar"
IdentityNumber: "454687864579"
Photograph: "C:\fakepath\background11.jpg"
Address: "Railway Station Delhi"
Locality: "Near NDLS Delhi"
City: "Delhi"
District: "Delhi"
State: "Delhi"
Country: "India"
Pincode: "475896"
Email: "rohan@gmail.com"
PhoneNumber: "4758965412"
__v: 0
```

Fig 4.12 Application data entry

Front End Implementation

From Now onwards we take our discussion to the next section where we discuss about the implementation of the frontend that How we develop the front end for this application.

There are of total 6 webpages contained in this project and form now onwards we are going to discuss each one of these one by one. These webpages are namely given below:

- Login or Signup or Home Page
- Application Window
- Application Preview
- Application Update
- About
- Contact Us

- **Login Page**

  Here we discuss about the implementation of the Login Page which we can say is the Sign Up or Home Page also. Here in the login page we use "div" to display the Navbar where the heading and the Several Navigation Links are present . For the Signup and Sign In Options there are several button which on click provide specific window to the user to interact with and do the further task. When user Click for the Sing in the sign in window appears there and when the user click for the sign up the Sign Up window appears on the screen. Whole the transmission is taking place with the help of JavaScript where we can actually dynamically change the webpage as per our need . all thanks to the **addEventListener()** function which make it possible to do this very operations.

  There is also a forgot button or link if the user forgot the details the user can able to click that very button and then after giving a valid email id he or she able to get the required info from the database.

  In the Homepage as the user enter the information for the signup all the info get stored in the database and a uniquely generated application number is given to the user for Login in future.

All these operations and processes taking place with the help of the axios which we are using to send Several Request to the server with the help of specific endpoints specified while creating the backend by providing the necessary information to them. Since we are doing all the operations asynchronously so here we are using the try and catch block to do all the asynchronous request so that we can able to rectify the errors whenever happened and can also show some error message with the help of alert() to the user.

There are two ways of entering into the application form page when user sign up he/she directly enter into the application page this only happen once only while sign up. But once coming out of the interface if user again want to go to the application page he/she need to enter the required credential while login.

The application number is one of the most important thing which we need to take the track of to the entire end of the process. Since we cannot actually transfer the JavaScript variable from one JavaScript file to one other JavaScript file in the “vanilla JavaScript” for this purpose we are using the concept of the session storage and local storage for proper functioning. And for navigation through the webpages we are using location.assign() and location.replace()function in the JavaScript.

The above is all about the logical implementation but one of the most important part of the website design is the look and feel or we can say the User Interface which matters a lot in the field of development. So for this very thing we are using the Concept Of Cascading Style Sheet (CSS) . This styling language give us a wide range of resources by which we can make our webpages attractive and can completely change the look and feel of the webpages. It has many layout Like Flex Box, Grid Layout etc. which make our task easy to implement what we actually think of to implement. A background is also provided to the webpage to make it beautiful. So, this is all about the login / signup / homepage now on further discussion we are going to discuss about the other pages.

- **Application Window**

Now after the login window we can move forward to the application window . This window is again consists of the Same portion of the header and navigation but here there is a horizontal bar which is used to display the application number of the applicants that is fetched from the session storage. There is also 2 button above that is Logout and Change Password button which the user can use to Logout from the window or to change the password respectively.

Now the all the left part of the application window is containing the HTML Form Which is distributed into several parts with the help of <legend> and <fieldset>. The <legend> tag will create a labelled outline around the container or the field inside the <fieldset>. And At last there is a checkbox and submit button to confirm the information with the help of checkbox in JavaScript and after clicking on the button the JavaScript will take the user to the preview page. As the user click the Submit Button all the data will get stored in the database with the help of the endpoint specified in the JavaScript there inside the try and catch block with the help of axios.

All the designing again is done with the help of the cascading Style Sheet (CSS). There is some specific border style mentioned in the CSS which is there in the webpage. All the things are under the Flex Box and all the flex box properties are used to align , justify or for doing anything productive with the webpages. A background is also provided to the webpage to make it beautiful.

All the buttons are integrated with some Transition and transform property so that when ever the user click or hover the button the user can able to see the required transition and transform in the webpage which is a great experience for the user.

All the checks are also integrated in the JavaScript so that any type of error can be taken care of with the help of warning that will be popped up to the user screen whenever user miss anything or do some errors. This is all about the application window.

- **Preview Window**

  Now after the application window we can move forward to the preview window. This window is again consists of the same portion of the header and navigation but here there is a horizontal bar which is used to display the application number of the applicants that is fetched from the session storage. There is also 2 button above that is Logout and Change Password button which the user can use to Logout from the window or to change the password respectively.

  Now the all the left part of the preview window is containing the HTML Form which is distributed into several parts with the help of <legend> and <fieldset>. The <legend> tag will create a labelled outline around the container or the field inside the <fieldset>. The HTML Forms here is auto filled with the data filled by the user at the time of filling the application form. And At last there is a checkbox and Edit Details and Print Pdf button which the user can use as per the need. If User select for the edit details button he/she directed to the edit details window and if the user select the Print The Pdf button the user can be able to save the entire screen as pdf for the future reference .As the user click the Edit details all the data from the database get loaded in to the edit details form where the user can able to edit the details with the help of the endpoint specified in the JavaScript there inside the try and catch block with the help of axios.

  All the designing again is done with the help of the cascading Style Sheet (CSS). There is some specific border style mentioned in the CSS which is there in the webpage. All the things are under the Flex Box and all the flex box properties are used to align , justify or for doing anything productive with the webpages. A background is also provided to the webpage to make it beautiful.

  All the buttons are integrated with some Transition and transform property so that when ever the user click or hover the button the user can able to see the required transition and transform in the webpage which is a great experience for the user.

  All the checks are also integrated in the JavaScript so that any type of error can be taken care of with the help of warning that will be popped up to the user screen whenever user miss anything or do some errors. This is all about the preview window.

- **Edit Details Window**

  Now after the preview window we can move forward to the Edit Details window. This window is again consists of the same portion of the header and navigation but here there is a horizontal bar which is used to display the application number of the applicants that is fetched from the session storage. There is also 2 button above that is Logout and Change Password button which the user can use to Logout from the window or to change the password respectively.

  Now the all the left part of the edit details window is containing the HTML Form which is distributed into several parts with the help of <legend> and <fieldset>. The <legend> tag will create a labelled outline around the container or the field inside the <fieldset>. The HTML Forms here is auto filled with the data filled by the user at the time of filling the application form. And At last there is a checkbox and submit details button which the user can use as per the need. When the user done with the updating the information which is provided by him/her earlier then he/she must check the checkbox and click the submit/update details button .As the user click the Submit/Update details all the data in the database get updated in to the edit details form and all the data get loaded in the preview window where the user can able to view the updated details with the help of the endpoint specified in the JavaScript there inside the try and catch block with the help of axios.

  All the designing again is done with the help of the cascading Style Sheet (CSS). There is some specific border style mentioned in the CSS which is there in the webpage. All the things are under the Flex Box and all the flex box properties are used to align , justify or for doing anything productive with the webpages. A background is also provided to the webpage to make it beautiful.

  All the buttons are integrated with some Transition and transform property so that whenever the user click or hover the button the user can able to see the required transition and transform in the webpage which is a great experience for the user.

  All the checks are also integrated in the JavaScript so that any type of error can be taken care of with the help of warning that will be popped up to the user screen whenever user miss anything or do some errors. This is all about the preview window.

- **About Window**
  This about window is just a normal single page static window where the brief description of what the project is mentioned in several points. Here all the content is in a single "DIV" and all the designing there is done with the help of CSS.

- **Contact Us Window**
  This about window is just a normal single page static window where some of the Contact Details are given for queries .Here all the content is in a single "DIV" and all the designing there is done with the help of CSS.

❖ There is 2 more layout or window which need to be discussed when talking about the front end that is forgot details window and change password window. Now we discuss these two dynamic window.

- **Change Password Window**: This window appear when user want to change the password at any point of time. This window is a absolute window having some border and here there are 2 entry boxes where the user can enter the new password and after clicking the change password button the new password get updated in the database with the help of the endpoint called using axioss under try catch block inside the JavaScript. There is also a cross Button which user can use to close the window. Again all the designing and transition are done with the help of Cascading Style Sheet (CSS).

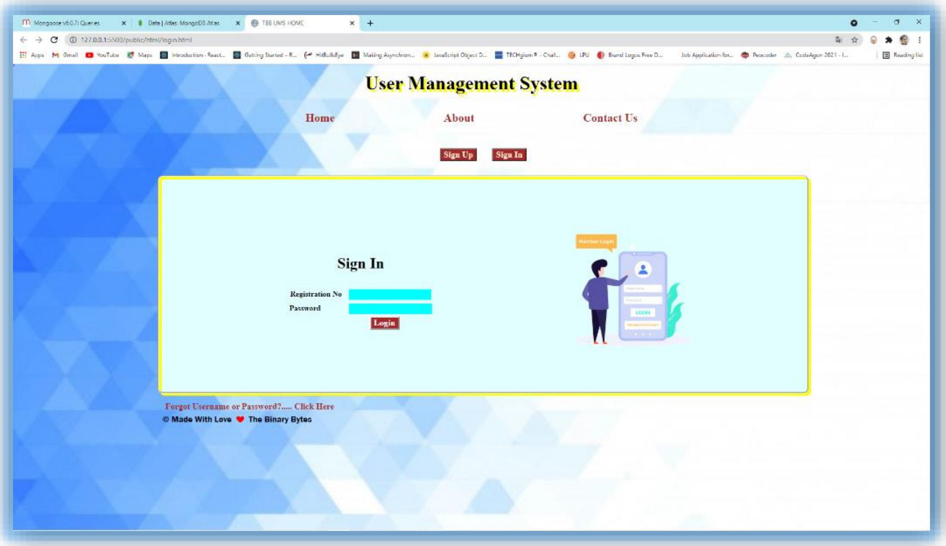
- **Forgot Details Window** : This window will appear when user want to forgot the details . This is a small square box window which will appear inside the Screen where there is a text box which ask the user for a valid email id , if the user give the right email id then it will fetch the info from the database with the help of endpoint and axios mentioned in the try catch block in the JavaScript and will display the details on the screen else it will show the warning message to the user. Again all the design is done with the help of Cascading Style Sheet (CSS).

Now the above is all the details of the front end and backend implementation. Since providing complete code in the report itself is not possible in practice and describing each and every small steps for the front end and backend is not possible here hence above is just the complete idea of how the things goes. If anyone is interested to saw actual code how thing goes. Then he/she will go to the git hub link provided for the whole project.

GitHub Repo Link

https://github.com/Sudeep-kr-1999/Application-Form-Management-System.git

Now below are the snaps of the interface how it looks in the frontend.



User Management System

Home About Contact Us

Sign Up Sign In

Sign In

Registration No

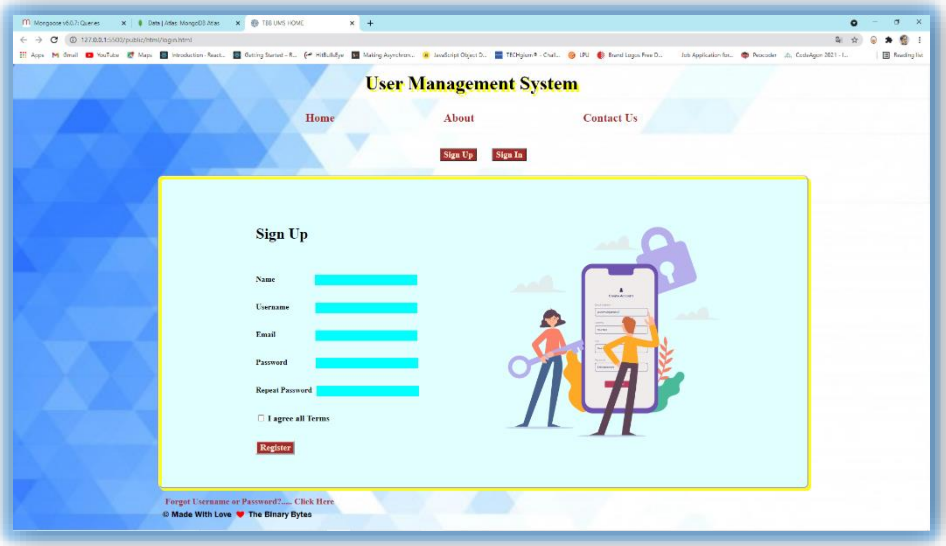
Password

Login

Forgot Username or Password?..... Click Here

© Made With Love ❤ The Binary Bytes

Fig 4.13 Sing In Window



User Management System

Home About Contact Us

Sign Up Sign In

Sign Up

Name

Username

Email

Password

Repeat Password

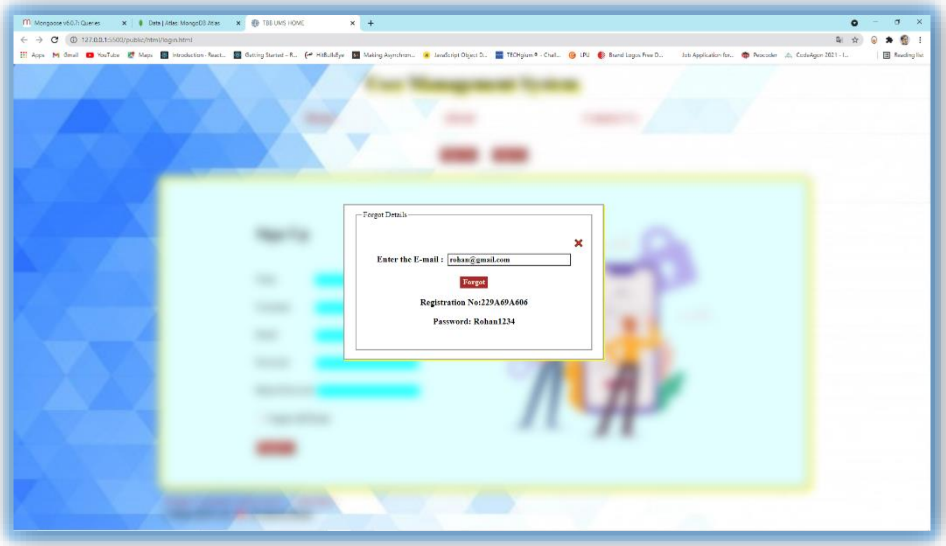
□ I agree all Terms

Register

Forgot Username or Password?..... Click Here

© Made With Love ❤ The Binary Bytes

Fig 4.14 Sin Up Window



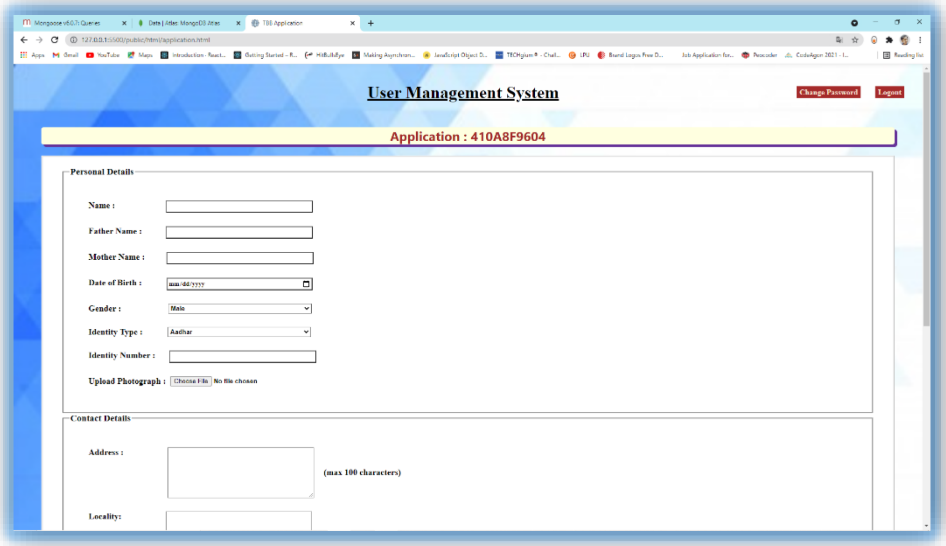
Forgot Details

Enter the E-mail : rohan@gmail.com

Registration No:229A69A606

Password: Rohan1234

Fig 4.15 Forgot Details Window



User Management System

Application : 410A8F9604

Personal Details

Name :

Father Name :

Mother Name :

Date of Birth : mm/dd/yyyy

Gender : Male

Identity Type : Aadhar

Identity Number :

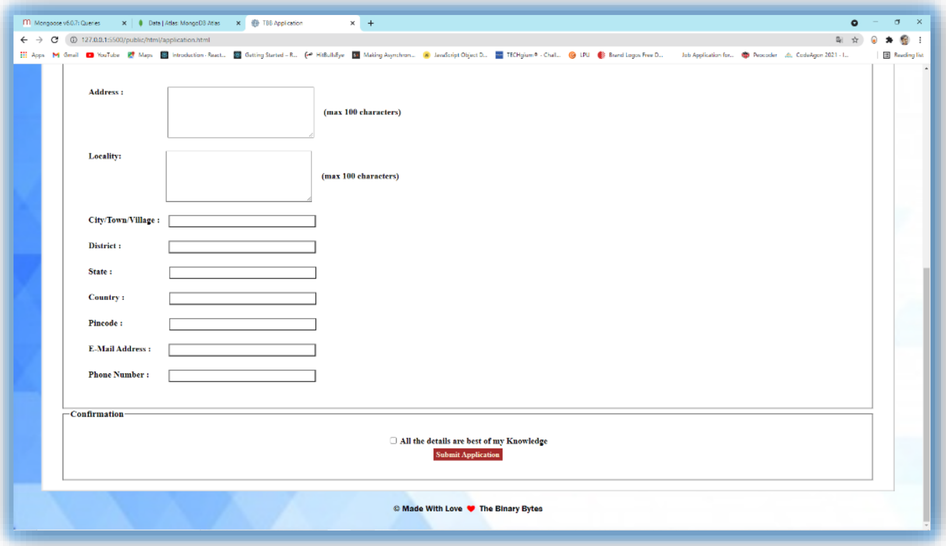
Upload Photograph : No file chosen

Contact Details

Address : (max 100 characters)

Locality:

Fig 4.16 Application Window-1


Mongoose v6.0.7: Queries
Data | Atlas: MongoDB Atlas
TBB Application
127.0.0.1:5500/public/html/application.html
Submit Application
© Made With Love ❤ The Binary Bytes

Address : (max 100 characters)

Locality: (max 100 characters)

City/Town/Village :

District :

State :

Country :

Pincode :

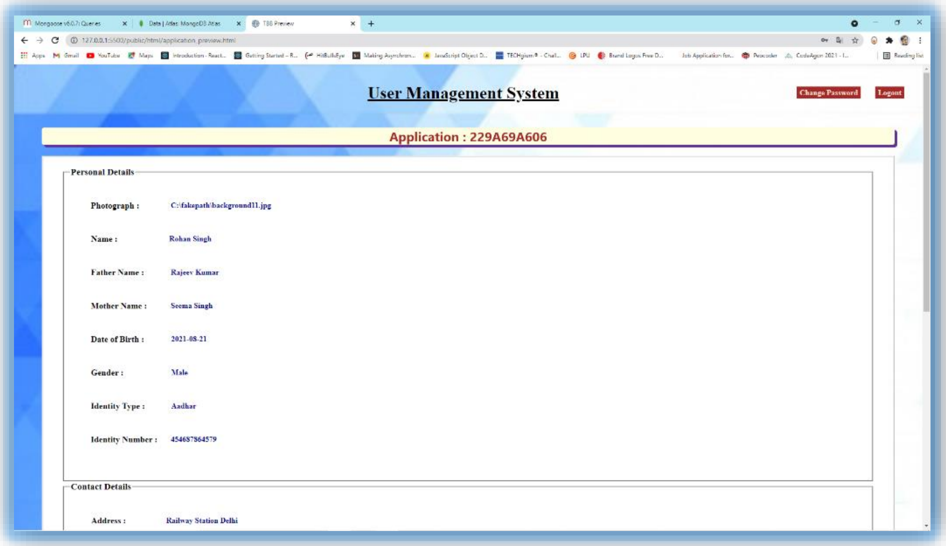
E-Mail Address :

Phone Number :

Confirmation

□ All the details are best of my Knowledge

Fig 4.17 Application Window 2


Mongoose v6.0.7: Queries
Data | Atlas: MongoDB Atlas
TBB Preview
127.0.0.1:5500/public/html/application_preview.html
Change Password
Logout

User Management System

Application : 229A69A606

Personal Details

Photograph : C:\fakepath\background11.jpg

Name : Rohan Singh

Father Name : Rajeev Kumar

Mother Name : Seema Singh

Date of Birth : 2021-08-21

Gender : Male

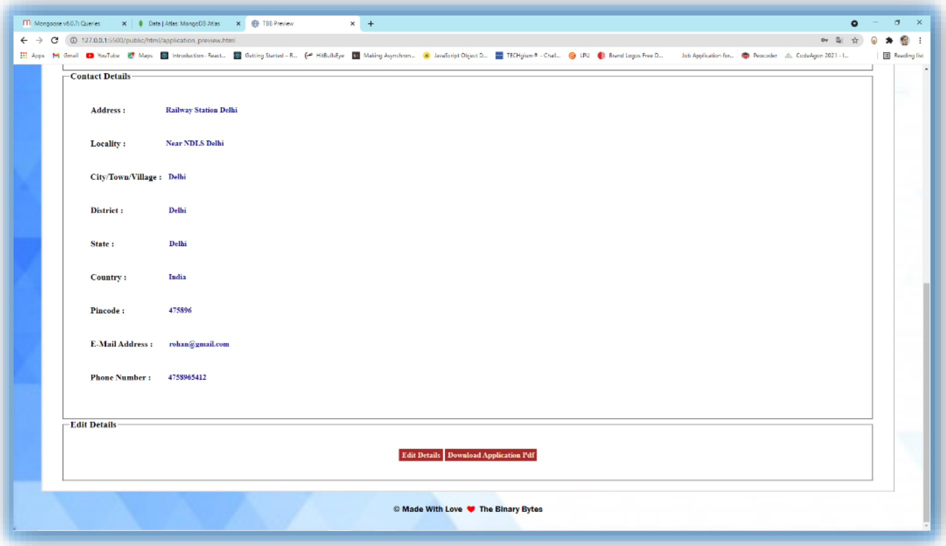
Identity Type : Aadhar

Identity Number : 454687864579

Contact Details

Address : Railway Station Delhi

Fig 4.18 Preview Window 1



Mongoose v6.0.7: Queries
Data | Atlas: MongoDB Atlas
TBB Preview
127.0.0.1:5500/public/html/application_preview.html
Edit Details
Edit Details
Download Application Pdf
© Made With Love ❤ The Binary Bytes

Contact Details

Address : Railway Station Delhi

Locality : Near NDLS Delhi

City/Town/Village : Delhi

District : Delhi

State : Delhi

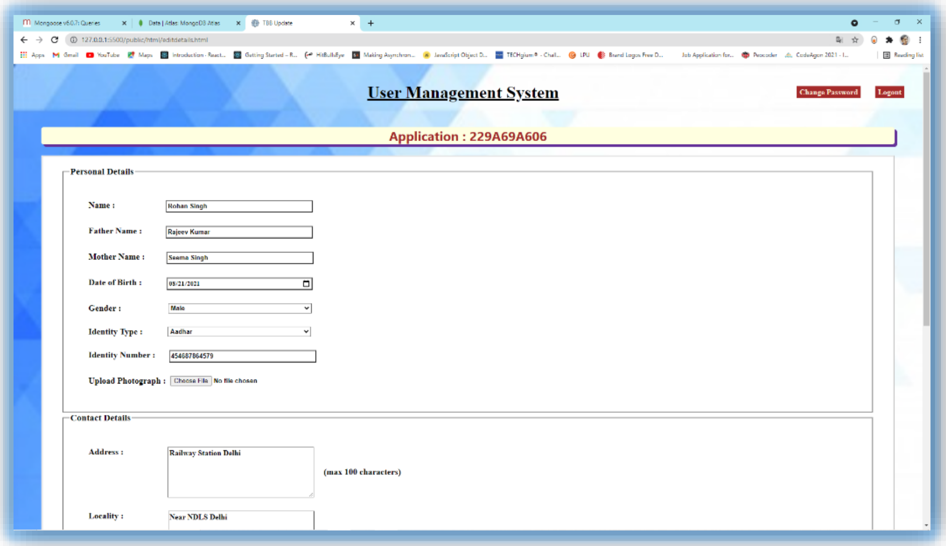
Country : India

Pincode : 475896

E-Mail Address : rohan@gmail.com

Phone Number : 4758965412

Fig 4.19 Preview Window 2



Application : 229A69A606

Personal Details

Name : Rohan Singh

Father Name : Rajeev Kumar

Mother Name : Seema Singh

Date of Birth : 09/21/2021

Gender : Male

Identity Type : Aadhar

Identity Number : 454687864579

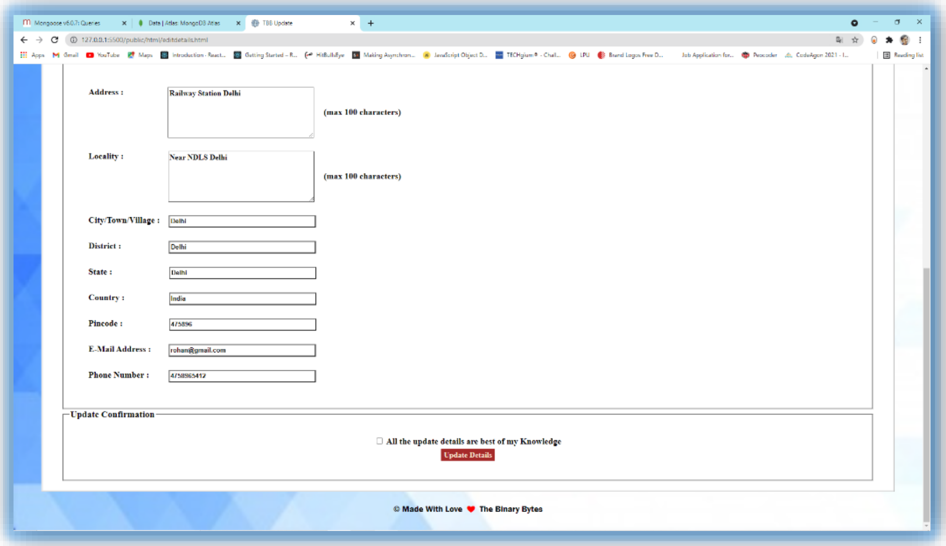
Upload Photograph : No file chosen

Contact Details

Address : Railway Station Delhi (max 100 characters)

Locality : Near NDLS Delhi

Fig 4.20 Update Window 1



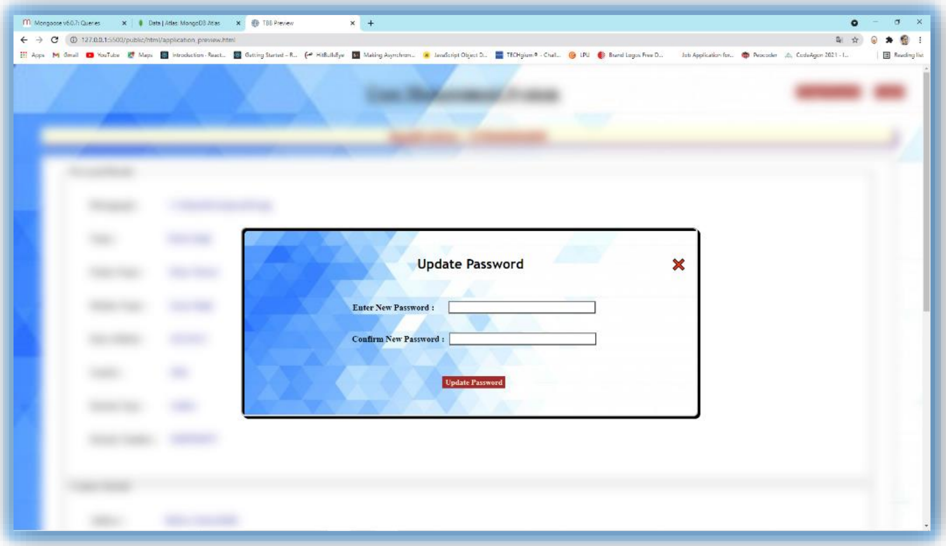
Address :	Railway Station Delhi	(max 100 characters)
Locality :	Near NDLS Delhi	(max 100 characters)
City/Town/Village :	Delhi	
District :	Delhi	
State :	Delhi	
Country :	India	
Pincode :	475896	
E-Mail Address :	rohan@gmail.com	
Phone Number :	4758965412	

Update Confirmation

□ All the update details are best of my Knowledge

Mongoose v6.0.7: Queries
Data | Atlas: MongoDB Atlas
TBB Update
127.0.0.1:5500/public/html/editdetails.html
Update Details
© Made With Love ❤ The Binary Bytes

Fig 4.21 Update Window 2



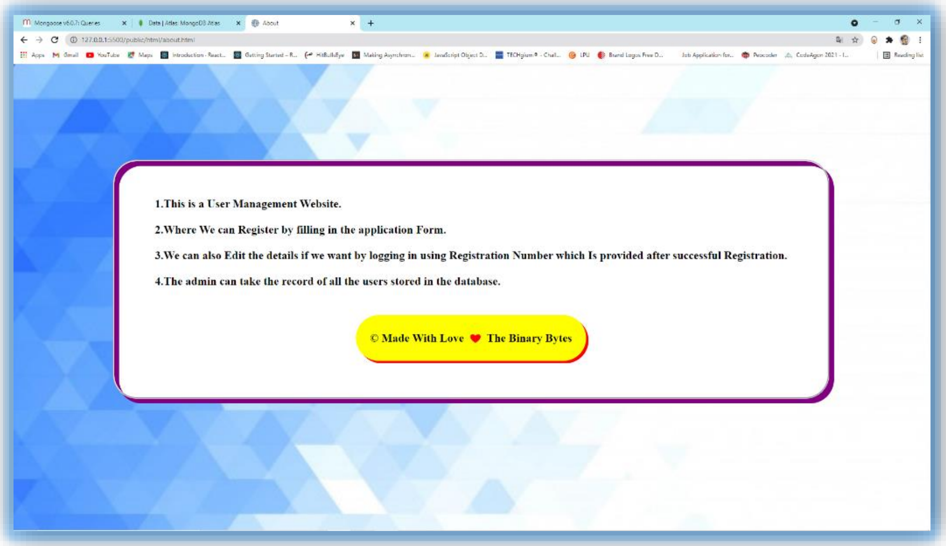
Update Password

Enter New Password :

Confirm New Password :

Mongoose v6.0.7: Queries
Data | Atlas: MongoDB Atlas
TBB Preview
127.0.0.1:5500/public/html/application_preview.html
Update Password

Fig 4.22 Change Password Window



1.This is a User Management Website.

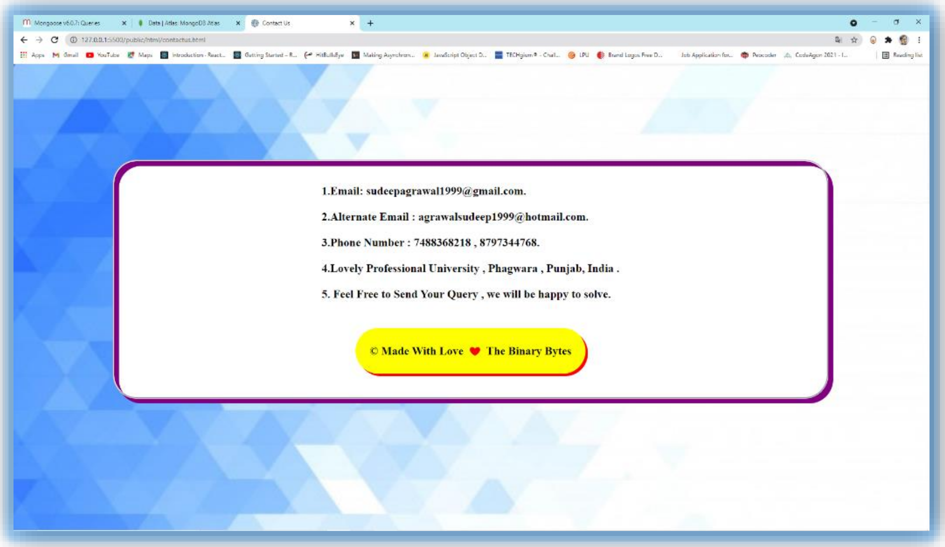
2.Where We can Register by filling in the application Form.

3.We can also Edit the details if we want by logging in using Registration Number which Is provided after successful Registration.

4.The admin can take the record of all the users stored in the database.

© Made With Love ❤ The Binary Bytes

Fig 4.23 About Window



1.Email: sudeepagrawal1999@gmail.com.

2.Alternate Email : agrawalsudeep1999@hotmail.com.

3.Phone Number : 7488368218 , 8797344768.

4.Lovely Professional University , Phagwara , Punjab, India .

5. Feel Free to Send Your Query , we will be happy to solve.

© Made With Love ❤ The Binary Bytes

Fig 4.24 Contact Us Window

Chapter 5

Learning Outcomes From Training

The most important part of for any person going under training of any technology , or any other subject is that how much information and lessons the person learnt from the training and with how much efficiency. This section deals with the new things and technologies that this whole project taught in course of this training.

As this project deals with most of the Logics of development field, then it is obvious that it deals with the technologies related to that. Hence below mention some of the technologies taught under this training.

- **HTML, CSS, JavaScript** : This training is about backend but only backend is not sufficient we must have a User Interface to interact with. This training gave the basic but applicable knowledge of the front end or we can say client side in the initial phases of the training period. Which lead as a helpful component for the designing of this project.

- **Node JS**: As the name say it all "Back end development with Node JS, Express and Mongo DB". Node JS is the important part of this training. This is very new topic and this training give us a good knowledge of the techniques involved in the Node JS to create backend and to do operation on the server Side. It taught us many NPM Modules used for this purpose like "http", "cors" etc. Also the Middleware that are very important to handle some important operation in the middle of the execution.

- **Express** : This training also taught us the Express Module which is very helpful for making REST API Request call to the server. It handle most of the thing smoothly without any big code or complexity. Without express we can still able to do all there but these logics require a complex implementation express make all the process easy. This is one of the reason it is in great demand now a days in the development field.

- **Mongo DB, MySQL** : This training also give us a good knowledge of the databases like MySQL and Mongo DB. This project work on Mongo DB because it is the new thing that this training taught and this project is the good way to implement and learn this technology. We may also use MySQL in place of it .

- And the most important thing this training taught is how to manage the time to learn new skills in our busy schedule that will be very helpful for all of us in the coming future.

CHAPTER 6

Progression Chart

This section deals with the visuals of continuous progression along with the training period. In things like training, internships or other areas where we learn something in a limited time span , the one of the most important things is to track our progress that is how much knowledge we are gaining day to day and out of that how much we are understanding and acquiring . This give us near exact idea that what is done and what needs to be done in the coming future.

One of the good way is to make a progression chart, this will help us to visualize the continuous progression. This chart can be on the weekly basis or on daily basis. Below is the progression chart which is on weekly basis which visualize the progression in the whole span of the training.

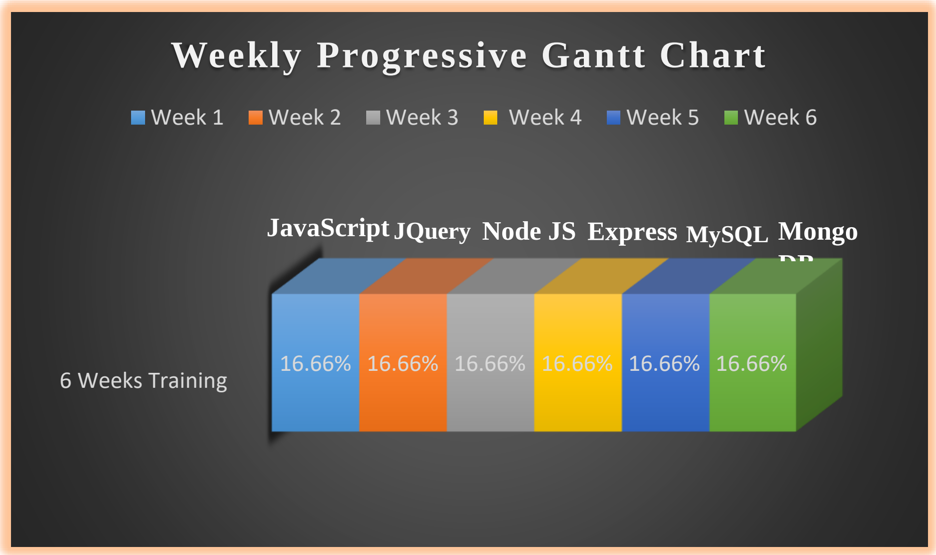


Fig 6.01 Weekly Progressive Gantt Chart

- **Project Legacy** : Talking about the project legacy, we can say that the legacy of any project is all about how informative the project is. On this line , and after proper discussion of the whole design and implementation of the project we can say that this project is very beneficial for us as it give us the glimpse of approximate all the technologies that one should learn to be a good developer, it teaches us the basics of the frontend and the backend logics that will be very helpful to build the big projects in the coming future.

CHAPTER 7 (FINAL CHAPTER)

CONCLUSION

This chapter is the final chapter of this report, this chapter is all about concluding all the things and technologies that we discuss in this report and providing a good ending review to the report. In conclusion of this report we can say that this project is a good way to learn the basic technologies related to the development on both backend and the frontend for one who is very new to this technology. These gives a complete glimpse and hand on practice so that one can able to understand the basic logics and can be able to expand it to the next level.

Its also a fact that One who is good at development may find this project easy but as it is correctly said all the big things always started as small things and by hard work they turned into big giants. Below are some of the points mentioned with which we are finally concluding this project.

- This project is based on the application management system which is one of the mostly used technology in the worldwide as no any field is there which is not using this technology.
- There are many real world examples we can talk about which are using this technology, So by this project we can able to understand the process that they are actually using in the backend to achieve what they want to achieve.
- This project indeed give us the basic knowledge related to front End via HTML, CSS, Java script and the Backend via Node JS and express which we can use it in the future for other good development purpose.
- This project also teach about some helpful technologies of Boot strap and j query which we can further use to make our website more interactive and beautiful.
- This project also give us the knowledge and application of the basic Database technologies like Mongo DB, MySQL which are widely used by many big giants companies as their database systems.

At last we can say that there is a lot of scope of the expansion of this project in coming future and we can actually add many new functionalities to it in future, with this statement we are concluding this project.

REFERENCES

This is the last part of this report and this is all about references. Below mentioned some of the references that are taken to make this project successful and form this project a thankful gesture to all the helpers references which help us make this project successful.

- https://Node JS.org/en/docs/
- https://expressjs.com/
- https://expressjs.com/en/starter/hello-world.html
- https://docs.mongo DB.com/
- https://docs.atlas.mongo DB.com/getting-started/
- https://mongoosejs.com/
- https://mongoosejs.com/docs/guide.html
- https://mongoosejs.com/docs/queries.html
- https://mongoosejs.com/docs/connections.html
- https://www.jotform.com/form-templates/category/application-form
- https://developer.mozilla.org/en-US/docs/Web/CSS
- https://www.w3schools.com/js/DEFAULT.asp
- https://JavaScript.info/
- https://www.freecodecamp.org/
- https://www.youtube.com/watch?v=RLtyhwFtXQA&list=PLWKjhJtqVAbmGQoa3vFjeRbRADAOC9drk
- https://www.youtube.com/watch?v=Oe421EPjeBE
- https://www.youtube.com/watch?v=TNV0_7QRDwY
- https://www.youtube.com/watch?v=jIsj0upCBAM&list=RDCMUCMZFwxv5l-XtKi693qMJptA&index=2