

A NOTES APP

Internship Project Report, Based on Web and MEAN Stack.

Objective :- Add a new note, Update existing note and Store them.

Conducted By :-

“Abhiyantrix And Sapience Academy Mysuru”

Under Supervision Of :

Mr. Krishna

Submitted By :

Arpitha B (1CC15CS001)

Chandan Y M (1CC15CS002)

Harshitha G E (1CC15CS005)

Meghana N (1CC15CS016)

Megha H R (1CC15CS014)

Vaibhav (1CC15CS026)

Shradda K (1CC15CS021)

Acknowledgement

We are pleased to acknowledge **Mr. Krishna** for his guidance during the course and commencing the project work.

We extend our sincere thanks to **Miss. Anjana Shashi Kiran** and **Mr. Harshith** who conducted this Internship program and helped us throughout the program.

We are also be thankful for all the members and team of “**Abhiyantrix And Sapience Academy Mysuru**”.

Contents

1. Acknowledgement
2. Contents
3. Introduction
4. Project Description
5. Source Code
6. Snapshots
7. References

Introduction

Overview

A Notes App, as the name indicates, this app is used to store Notes. This app helps user to Login or Sign-Up to the app and add a new note or update the existing note in his/her account.

This project has been conducted using HTML and CSS. Hypertext Markup Language (HTML) which is a leading language that is being used in Web Development. Cascading Style Sheet (CSS) is a style sheet used to apply styles and formatting the page which makes the Web Page looks attractive and effective. MEAN Stack is used for backend connection and creating a database to store the user details and user notes.

Project Description

The project's frontend has been edited using Notepad++ text editor. MEAN (MongoDB, Express, Angular and Nodes) is used to connect the frontend with the backend. MongoDB is used for the database, to store user name, email, password, and notes that user added to the account. The project uses several steps, which is as follows:

1. If the user already registered to app, then it asks for User Login. If user is new to app, then it allows user to Sign-Up to the app and then Login to the app to go to the notes page.
2. The user can add or edit or delete Notes and save changes to the app.
3. Later the app stores the user Notes to the database, so that user can open it whenever user needs to.

Source Code

Code for Login page:

```
<html>
  <head>
    <title>Login</title>
    <link rel="stylesheet" href="login.css">
      <script src="http://code.jquery.com/jquery-1.11.1.min.js"></script>
      <script src = "login.js" type="text/javascript"></script>
  </head>
  <body>
    <div class="login-box">
      <form>
        <h1>Login</h1>
        <div class="textbox">
          <i class="fa fa-envelope" aria-hidden="true"></i>
          <input type="text" placeholder="Email Id" name="" value="">
        </div>
        <div class="textbox">
          <i class="fa fa-lock" aria-hidden="true"></i>
          <input type="password" placeholder="Password" name="" value="">
        </div>
        <a href="notes.html">
          <input class="btn" type="button" name="" value="Sign In">
        </a><br><br>
        Create an Account?<a href="signup.html">Click Here!</a>
      </form>
    </div>
  </body>
</html>
```

Code for Sign-Up page:

```
<html>
  <head>
    <title>Sign Up</title>
    <link rel="stylesheet" href="login.css">
      <script src="http://code.jquery.com/jquery-1.11.1.min.js"></script>
      <script src = "login.js" type="text/javascript"></script>
  </head>
```

```

<body>
  <div class="login-box">
    <form>
      <h1>Sign Up</h1>
      <div class="textbox">
        <i class="fa fa-user" aria-hidden="true"></i>
        <input type="text" placeholder="First Name" name="" value="">
      </div>
      <div class="textbox">
        <i class="fa fa-user" aria-hidden="true"></i>
        <input type="text" placeholder="Last Name" name="" value="">
      </div>
      <div class="textbox">
        <i class="fa fa-envelope" aria-hidden="true"></i>
        <input type="text" placeholder="Email id" name="" value="">
      </div>
      <div class="textbox">
        <i class="fa fa-phone" aria-hidden="true"></i>
        <input type="text" placeholder="Phone Number" name="" value="">
      </div>
      <div class="textbox">
        <i class="fa fa-lock" aria-hidden="true"></i>
        <input type="password" placeholder="Password" name="" value="">
      </div>
      <a href="notes.html">
        <input class="btn" type="button" name="" value="Sign Up">
      </a>
    </form>
  </div>
</body>
</html>

```

Code for Notes page:

```

<html>
<head>
<title>NOTES</title>
<script src="http://code.jquery.com/jquery-1.11.1.min.js"></script>
<script src = "login.js" type="text/javascript"></script>
</head>
<body style="background-color:powderblue;">
<pre> </pre>
<pre> </pre>
<form><h1 style="font-family:verdana;">

```

```

<pre><font color="black"><center> Enter your notes here:<input type="text"  name="notes"
size="90"></center></font></pre></h1>
<br>
<center><input  type="button"  name="save"  value="save"  style="height: 30px;
width:50px;left: 50; top: 50;">
<input type="button" name="undo" value="undo" style="height: 30px; width:50px;left: 50;
top: 50;">
<input type="button" name="open" value="open" style="height: 30px; width:50px;left: 50;
top: 50;">
</center>
</form>
</body>
</html>

```

Code for server connection:

```

const express = require('express');
const morgan = require('morgan');
const mongoose = require('mongoose');
const bodyParser = require('body-parser');

const routes = require('./routes/routes');

mongoose.Promise = global.Promise;

const app = express();

mongoose.connect('mongodb://localhost:27017/user',{ useNewUrlParser: true });

app.use(bodyParser.urlencoded({ extended: true }));
app.use(morgan('dev'));

routes(app);

app.use((err,req,res,next) => {
    res.status(422).send({ status: false, error: err.message });
});

const ip = '127.0.0.1';

app.listen(3000,ip, () => {
    console.log('Server listening at '+ip+'!3000');
});

```

Code for Login and Sign-Up:

```
const user = require('../controller/user');
const path = require('path');

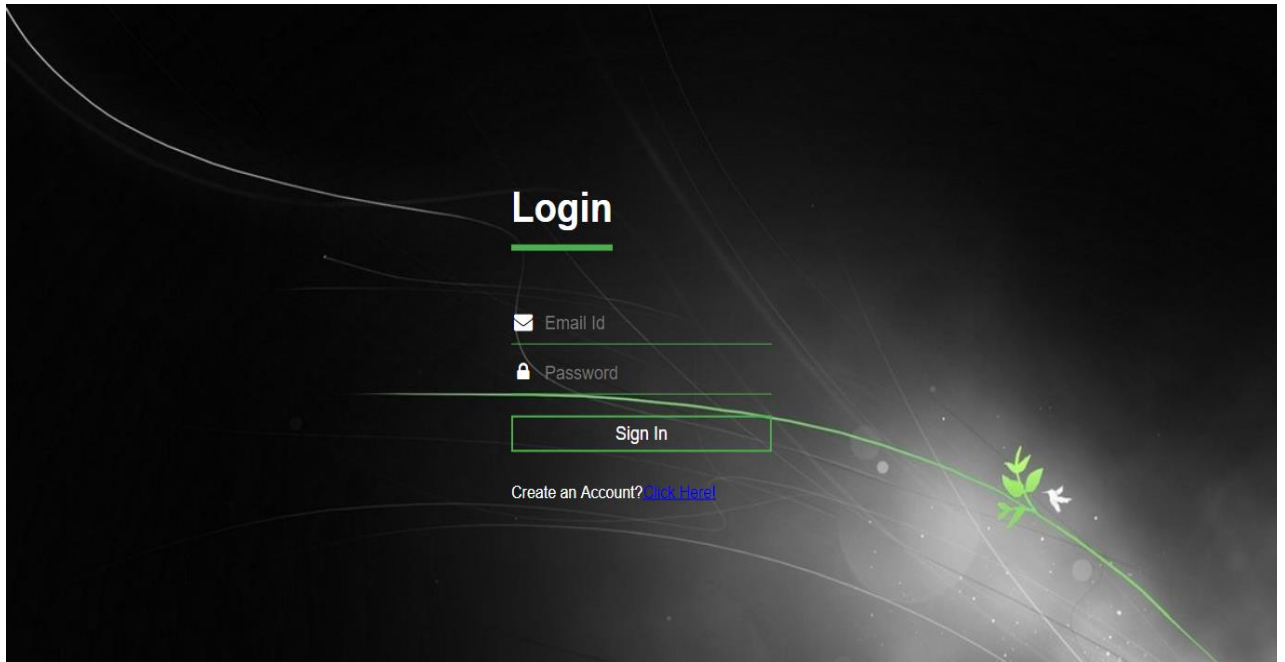
module.exports = (app) => {
  app.get('/', (req,res) => {
    res.sendFile(path.join(__dirname+'../login.html'));
  });

  app.post('/login',user.login);

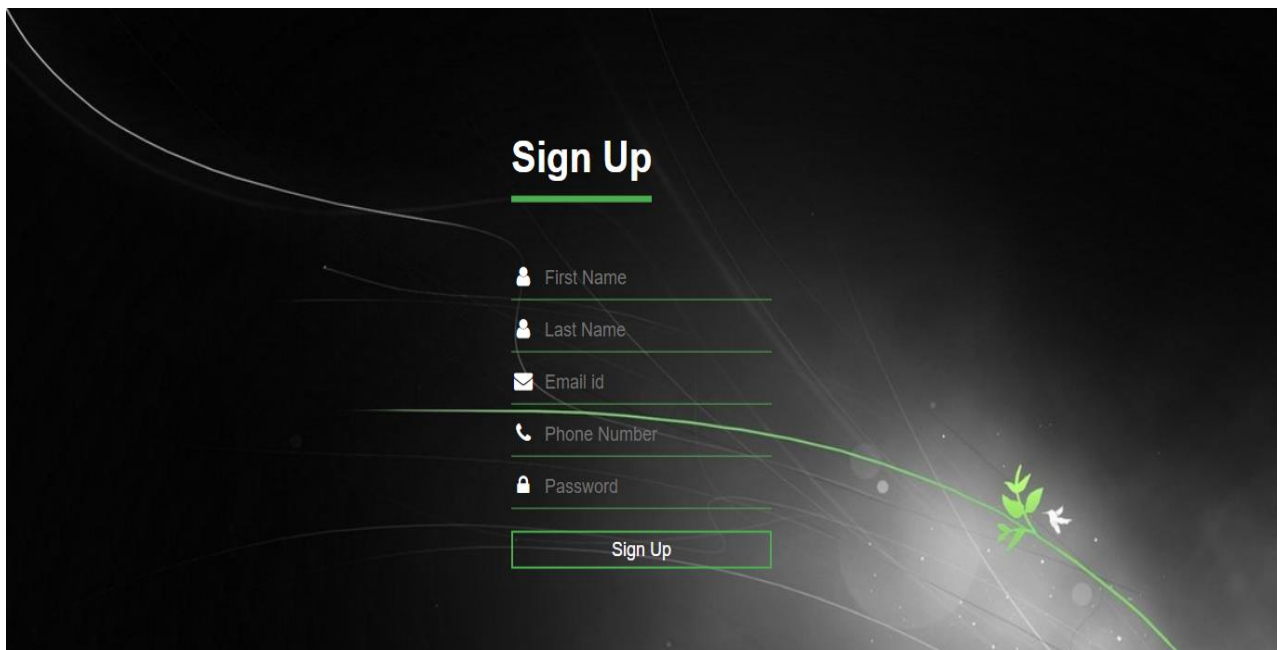
  app.get('/signup', (req,res) => {
    res.sendFile(path.join(__dirname+'../signup.html'));
  });

  app.post('/signup',user.signup);
}
```


Snapshots



This snapshot shows the page that a user can Login to the app.



This snapshot shows the page that a user can Sign-Up for the app.

Enter your notes here:

This snapshot shows the page that a user can add or update notes to his account in the app.

References

1. YouTube videos on MEAN stack.
2. HTML & CSS The Complete Reference, Fifth edition, Thomas A. Powell