Markdown Editor - MarkIt

Project Report

Fall Semester 2022-2023

Project By: -

Harsh Wardhan	20BCI0051
Prathamesh Sudhakar	20BCI0123
Preet Aryan Gupta	20BCI0165

In partial fulfilment for the award of the degree of

B. Tech

in

Computer Science and Engineering

Under The Guidance of

PROF. JAYAKUMAR K

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

Vellore-632014, Tamil Nadu, India

School of Computer Science and Engineering

Introduction

Note taking is a crucial human activity that allows us to document and reproduce knowledge in an accessible form. Writing down everything you hear or read is not the only way to take notes. It involves reviewing, connecting, and synthesizing concepts from your lectures and/or reading.
Making notes facilitates:

- Engaging actively in your lectures, reading, and study sessions.
- Comprehending the material and elucidating your thinking
- selectively identifying essential ideas
- Remember information, organize thoughts and make connections

It is equally important to have all your notes be easily accessible and easy to comprehend. Mark-it allows users to make well formatted documents easily and conveniently without ever needing to lift your hands from the keyboard. All your notes are saved securely over the internet and can be accessed from anywhere and any device as long as you have a stable internet connection. Mark-it uses markdown which is not only an incredibly powerful language to format notes, it is also very straightforward and easy to grasp.

Markdown

Markdown is a lightweight markup language used to format plaintext text documents. Markdown, which was created by John Gruber in 2004, is now one of the most popular markup languages in the world.
One can use standard markdown formatting elements or HTML elements to style their document as per requirement.

Abstract

Markit is a note-taking web app that makes sure you keep your notes handy and organized. Markit uses Markdown for easy and powerful formatting allowing users to make incredible notes with ease. These notes use a standardized format and hence the notes you make can be transferred over to any other markdown viewer. It stores your notes securely over the web, making it accessible from anywhere across the globe. Standard markdown notes can be generated and stored with ease. Markit is a note-taking web-app that makes sure you keep your notes handy and organized. Markit uses Markdown for easy and powerful formatting, allowing users to make incredible notes with ease. It stores your notes securely on the web using mysql, making it accessible from anywhere across the globe. We use cryptographic techniques to store your personal details.

Problem Statement

There are a lot of note making tools available on the internet that are either too heavy or not powerful enough. Most note-taking apps also use proprietary file formats to save files, which can only be viewed on those particular softwares. Mark-it uses a markdown format that is universally accepted and allows you to share your notes easily while being light on your system. Your notes are backed up on the web securely and allow you to access them from anywhere.

Technical Specification

HTML and CSS is used for designing the web pages.The home page,login, signup page and the notes page is designed using the html and css.The validation of the username ,password taken from the forms is done using javascript, node js.The notes page is managed using node js,express js and is connected to the database using MySQL where tables are created to store the information.

Existing system problems

Allows users to make only one note at a time while in MarkIt users can make 4 notes.
Formatting the note in another system makes the user search the menu and then do the changes whereas this editor allows the user to make changes in the textarea itself by just using some elements.
For the existing system the note used to get shared as a text.Here the note is saved in pdf format which makes it easier to share the file.

Proposed System Design

List of Modules

1. Registration page (Signup Page)
2. Login Page
3. Notes Page
4. Marked Parser
5. Edit Button
6. Sharing

Modules Description

1. **Registration Page:** This page contains a form for a new user to sign up to the site. Using the email address as a username a new account is created and password is being set for the account. HTML and CSS is used for designing the page and for the validation of the information javascript is used if the email id entered is valid or not and the password fulfils the requirements.
2. **Login Page:** This page contains the form for logining in to the website which asks for the username and password. It also has the forgot password and signup links which redirects to the respective pages.HTML and CSS is used for designing the page and for the validation of the information javascript is used checks if the email id is valid and password entered matches the requirements.
3. **Notes Page:**This page contains a textarea where users can write the notes. HTML, CSS, Javascript, Node js is used. You can make notes, edit them and share them.
4. **Marked Parser:** The note the user makes is transformed into HTML script using markdown. This helps the user to make notes with style even when the user does not have the time to make the design for the notes. This can be done only by adding certain symbols in the text.
5. **Edit Button:** A user can edit the note whenever he wants to change it without hassle. The data is edited and stored securely in our MySQL server and is retrieved when the user logs in again in their account.
6. **Sharing:** The notes written and saved can be shared using the share button at the right of the note textarea.This option allows the user to save the notes in the pdf format which can be saved in a folder chosen by the user.After saving the file it can be shared through various applications.which should be carried out for better user experience.

DATABASE

Database Tables:

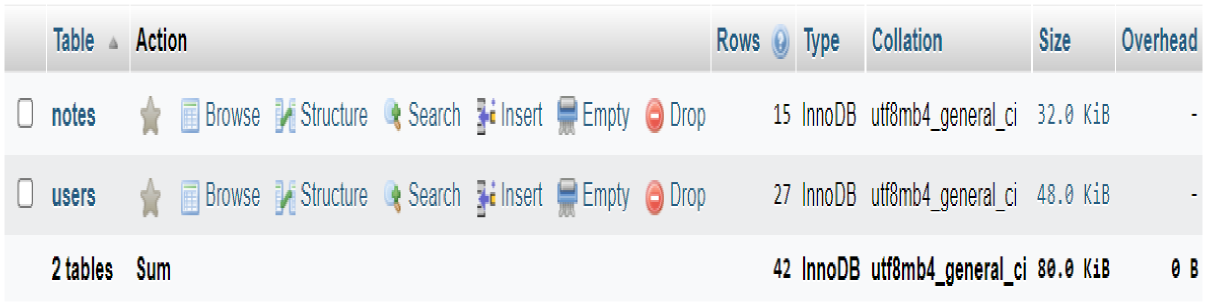
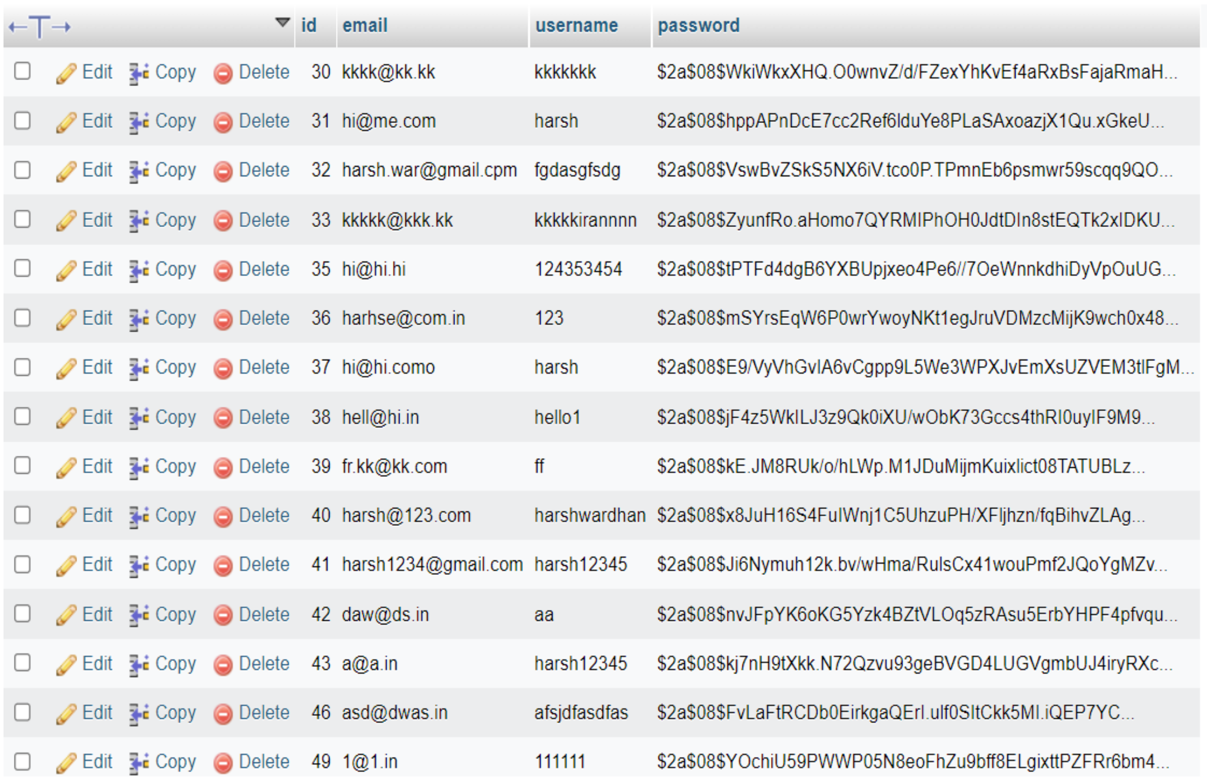


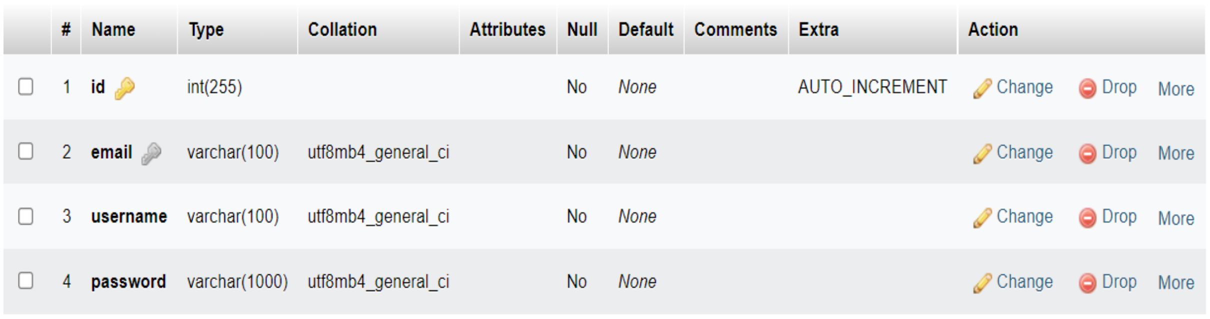
	Table	Action	Rows	Type	Collation	Size	Overhead
☐	notes	Browse Structure Search Insert Empty Drop	15	InnoDB	utf8mb4_general_ci	32.0 KiB	-
☐	users	Browse Structure Search Insert Empty Drop	27	InnoDB	utf8mb4_general_ci	48.0 KiB	-
	2 tables	Sum	42	InnoDB	utf8mb4_general_ci	80.0 KiB	0 B

Users table:

```
CREATE TABLE `users` (
 `id` int(255) NOT NULL AUTO_INCREMENT,
 `email` varchar(100) NOT NULL,
 `username` varchar(100) NOT NULL,
 `password` varchar(1000) NOT NULL,
 PRIMARY KEY (`id`),
 UNIQUE KEY `email` (`email`)
) ENGINE=InnoDB AUTO_INCREMENT=8 DEFAULT CHARSET=utf8mb4
```



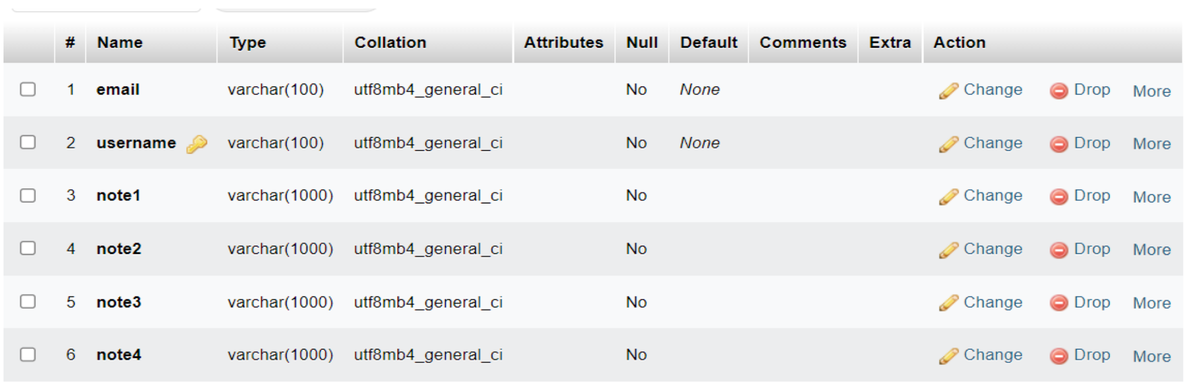
←T→				id	email	username	password
☐	Edit	Copy	Delete	30	kkkk@kk.kk	kkkkkkk	$2a$08$WkiWkxXHQ.O0wnvZ/d/FZexYhKvEf4aRxBsFajaRmaH...
☐	Edit	Copy	Delete	31	hi@me.com	harsh	$2a$08$hppAPnDcE7cc2Ref6lduYe8PLaSAxoazjX1Qu.xGkeU...
☐	Edit	Copy	Delete	32	harsh.war@gmail.cpm	fgdasgfsdg	$2a$08$VswBvZSkS5NX6iV.tco0P.TPmnEb6psmwr59scqq9QO...
☐	Edit	Copy	Delete	33	kkkkk@kkk.kk	kkkkkirannnn	$2a$08$ZyunfRo.aHomo7QYRMIPhOH0JdtDIn8stEQTk2xIDKU...
☐	Edit	Copy	Delete	35	hi@hi.hi	124353454	$2a$08$tPTFd4dgB6YXBUpjxeo4Pe6//7OeWnnkdhiDyVpOuUG...
☐	Edit	Copy	Delete	36	harhse@com.in	123	$2a$08$mSYrsEqW6P0wrYwoyNKt1egJruVDMzcMijK9wch0x48...
☐	Edit	Copy	Delete	37	hi@hi.como	harsh	$2a$08$E9/VyVhGvlA6vCgpp9L5We3WPXJvEmXsUZVEM3tlFgM...
☐	Edit	Copy	Delete	38	hell@hi.in	hello1	$2a$08$jF4z5WklLJ3z9Qk0iXU/wObK73Gccs4thRI0uylF9M9...
☐	Edit	Copy	Delete	39	fr.kk@kk.com	ff	$2a$08$kE.JM8RUk/o/hLWp.M1JDuMijmKuixlict08TATUBLz...
☐	Edit	Copy	Delete	40	harsh@123.com	harshwardhan	$2a$08$x8JuH16S4FulWnj1C5UhzuPH/XFljhzn/fqBihvZLAg...
☐	Edit	Copy	Delete	41	harsh1234@gmail.com	harsh12345	$2a$08$Ji6Nymuh12k.bv/wHma/RulsCx41wouPmf2JQoYgMZv...
☐	Edit	Copy	Delete	42	daw@ds.in	aa	$2a$08$nvJFpYK6oKG5Yzk4BZtVLOq5zRAsu5ErbYHPF4pfvqu...
☐	Edit	Copy	Delete	43	a@a.in	harsh12345	$2a$08$kj7nH9tXkk.N72Qzvu93geBVGD4LUGVgmbUJ4iryRXc...
☐	Edit	Copy	Delete	46	asd@dwas.in	afsjdfasdfas	$2a$08$FvLaFtRCDb0EirkgaQErl.ulf0SltCkk5Ml.iQEP7YC...
☐	Edit	Copy	Delete	49	1@1.in	111111	$2a$08$YOchiU59PWWP05N8eoFhZu9bff8ELgixttPZFRr6bm4...



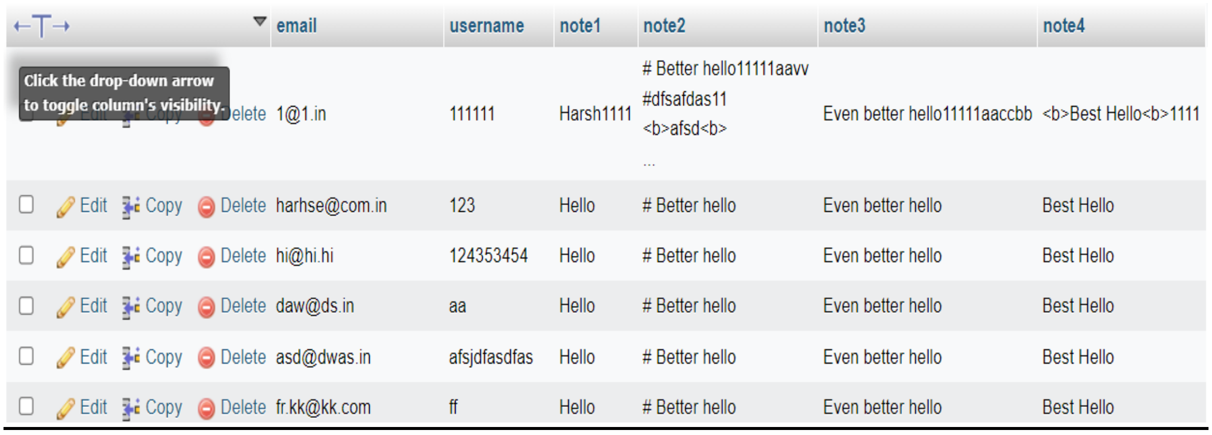
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(255)			No	None		AUTO_INCREMENT	Change Drop More
2	email	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More
3	username	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More
4	password	varchar(1000)	utf8mb4_general_ci		No	None			Change Drop More

Notes table:

```
CREATE TABLE `notes` (
 `username` varchar(100) NOT NULL,
`email` varchar(100) NOT NULL,
 `note1` varchar(1000) NOT NULL,
`note2` varchar(1000) NOT NULL,
`note3` varchar(1000) NOT NULL,
`note4` varchar(1000) NOT NULL,
 UNIQUE KEY `username` (`username`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
```



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	email	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More
2	username	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More
3	note1	varchar(1000)	utf8mb4_general_ci		No				Change Drop More
4	note2	varchar(1000)	utf8mb4_general_ci		No				Change Drop More
5	note3	varchar(1000)	utf8mb4_general_ci		No				Change Drop More
6	note4	varchar(1000)	utf8mb4_general_ci		No				Change Drop More

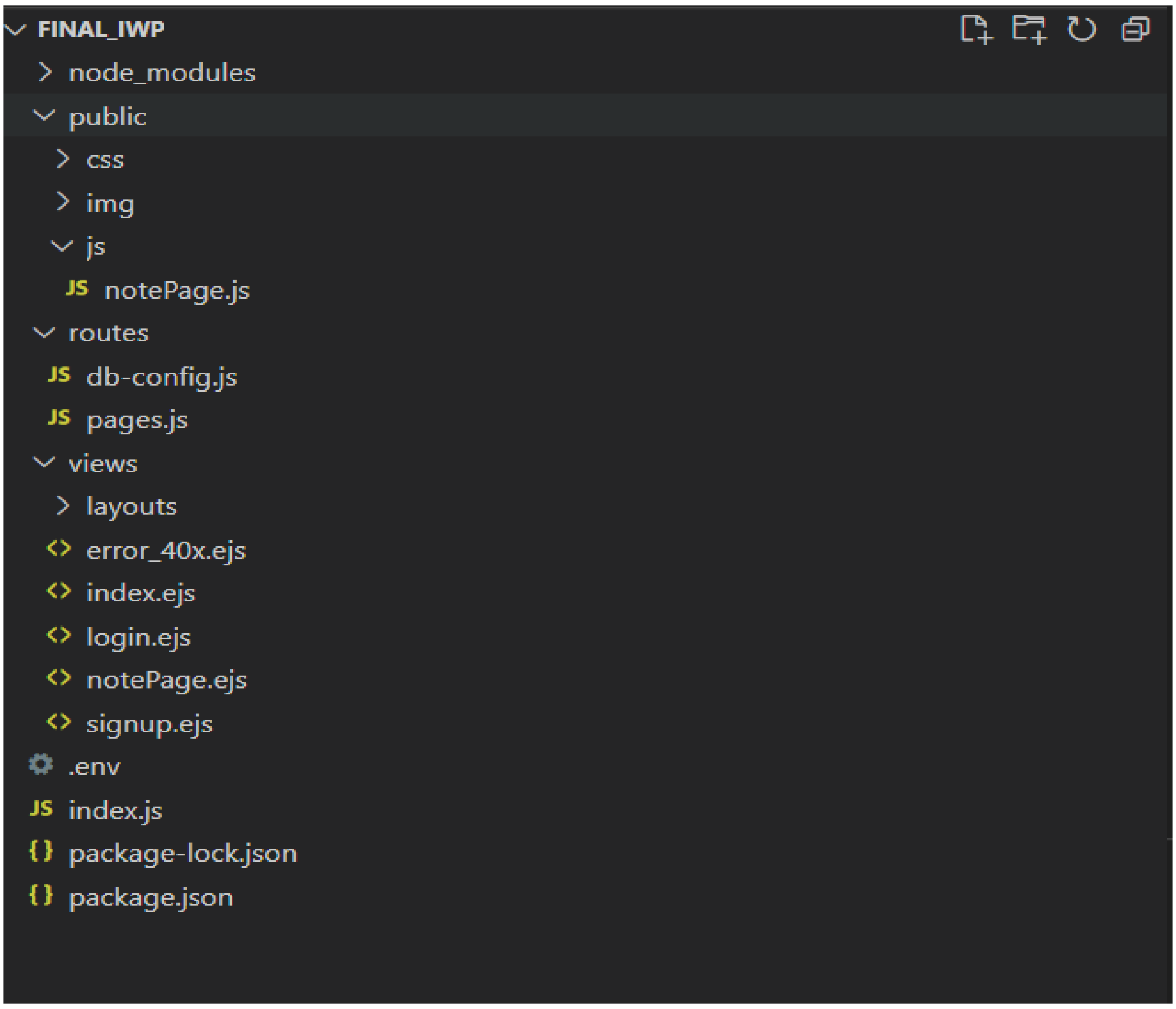


Click the drop-down arrow to toggle column's visibility.

←T→	email	username	note1	note2	note3	note4
Delete	1@1.in	111111	Harsh1111	# Better hello11111aavv #dfsafdas11 <b>afsd<b> ...	Even better hello11111aaccbb	<b>Best Hello<b>1111
Edit Copy Delete	harhse@com.in	123	Hello	# Better hello	Even better hello	Best Hello
Edit Copy Delete	hi@hi.hi	124353454	Hello	# Better hello	Even better hello	Best Hello
Edit Copy Delete	daw@ds.in	aa	Hello	# Better hello	Even better hello	Best Hello
Edit Copy Delete	asd@dwas.in	afsjdfasdfas	Hello	# Better hello	Even better hello	Best Hello
Edit Copy Delete	fr.kk@kk.com	ff	Hello	# Better hello	Even better hello	Best Hello

CODE:

The Project File Structure


```
FINAL_IWP
  > node_modules
  v public
    > css
    > img
    v js
      notePage.js
  v routes
    db-config.js
    pages.js
  v views
    > layouts
    error_40x.ejs
    index.ejs
    login.ejs
    notePage.ejs
    signup.ejs
  .env
  index.js
  package-lock.json
  package.json
```

Index.ejs:

```
const express = require("express");
const db = require("./routes/db-config");
const app = express();
require("dotenv").config();

const createHttpError = require('http-errors');
const morgan = require('morgan');
app.use(morgan('dev'));
const session = require('express-session');
const connectFlask = require('connect-flash');

const PORT = process.env.PORT || 5000;
app.use("/js", express.static(__dirname + "/public/js"));
app.use("/css", express.static(__dirname + "/public/css"));
app.use("/img", express.static(__dirname + "/public/img"));

app.set("view engine", "ejs");
app.set("views", "./views");
app.use(express.json());
app.use(express.urlencoded({ extended: false }));




app.use(session({
   secret: "some secret message",
   resave: false,
   saveUninitialized: false,
   cookie: {
      // secure:true,
      httpOnly: true
   }
})
);
app.use(connectFlask());

app.use('/', require("./routes/pages"));
// app.use("/api", require("./controllers/auth"));
// app.use('/', require('./routes/index.route'));
// app.use('/auth', require('./routes/auth.route'));
// app.use('/user', require('./routes/user.route'));

app.use((req, res, next) => {
   next(createHttpError.NotFound());
});
app.use((error, req, res, next) => {
   error.status = error.status || 500;
   res.status(error.status);
   res.send(error);
})
app.listen(PORT);
```

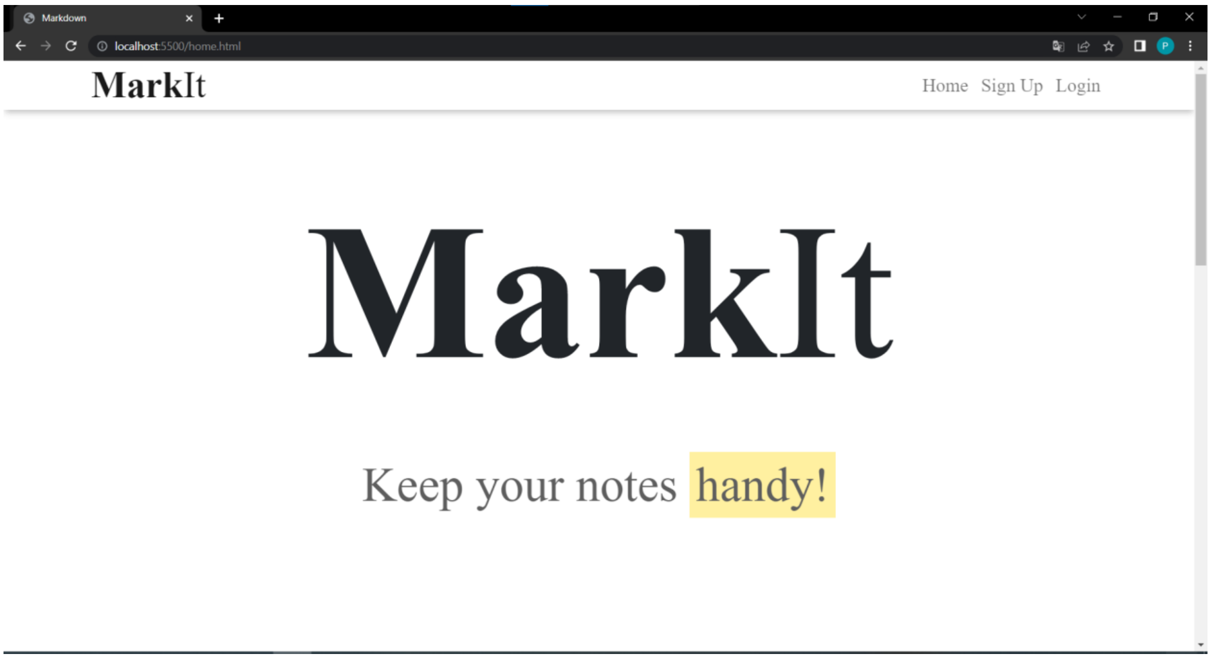
Homepage:

```
<!DOCTYPE html>
<html lang="en">

<head>
   <meta charset="utf-8">
   <meta name="viewport" content="width=device-width, initial-scale=1">

   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0/dist/css/bootstrap.min.css"
rel="stylesheet"
      integrity="sha384-
gH2yIJqKdNHPEq0n4Mqa/HGKIhSkIHeL5AyhkYV8i59U5AR6csBvApHHNl/vI1Bx"
crossorigin="anonymous">
   <!-- <link rel="stylesheet" href="css/bootstrap.min.css"> -->
   <link rel="preconnect" href="https://fonts.googleapis.com">
```

```html
    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
    <link
href="https://fonts.googleapis.com/css2?family=Be+Vietnam+Pro:wght@100;700&family=Lora
&display=swap"
        rel="stylesheet">
    <link rel="stylesheet" href="css/home.css">
    <title>Markdown</title>
</head>

<body>

    <nav class="navbar navbar-light navbar-expand-md bg-light justify-content-center sticky-
top">
        <div class="container">
            <a href="/" class="navbar-brand d-flex me-auto">Mark<span>It</span></a>
            <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#collapsingNavbar3">
                <span class="navbar-toggler-icon"></span>
            </button>
            <div class="navbar-collapse collapse w-100" id="collapsingNavbar3">
                <ul class="nav navbar-nav ms-auto w-100 justify-content-end">
                    <li class="nav-item">
                        <a class="nav-link" href="/">Home</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="/signup">Sign Up</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="/login">Login</a>
                    </li>
                </ul>
            </div>
        </div>
    </nav>
    <section class="hero">
        <div class="container mt-5">
            <div class="row d-flex justify-content-center align-items-center">
                <div class=" project">
                    <p class="name">Mark<span>It</span></p>
                    <p class="details">Keep your notes <span>handy!</span></p>
                </div>
            </div>
        </div>
    </section>

    <section class="photo">
```

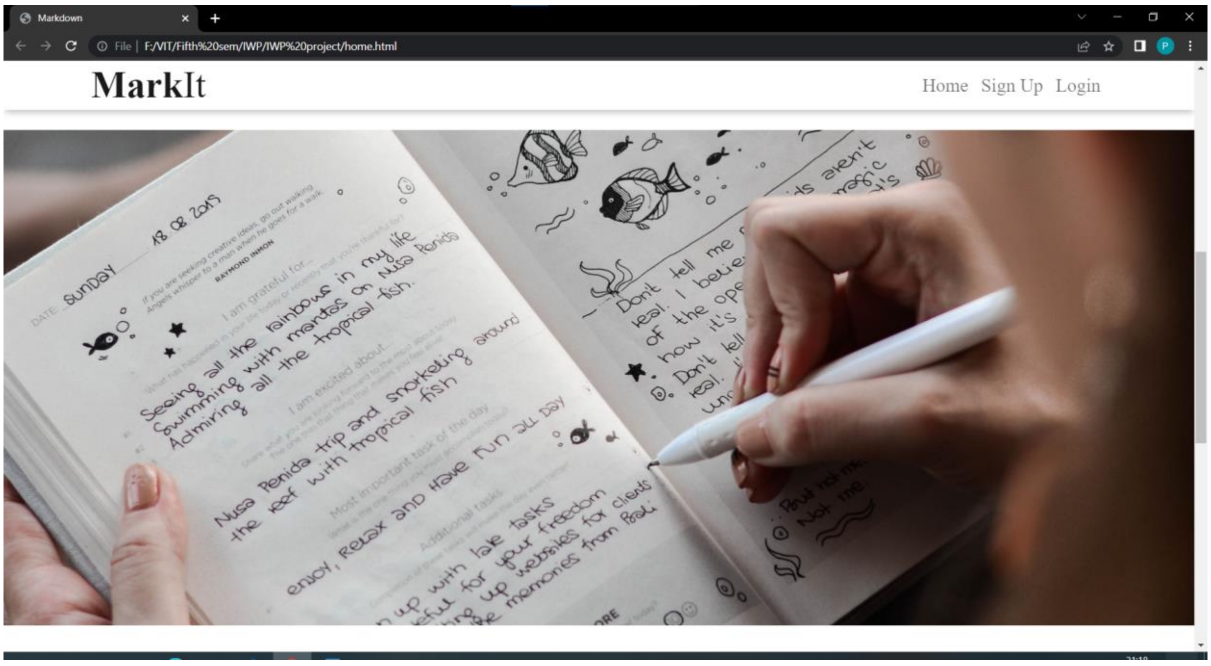
```html
        <div class="container-fluid">
            <div class="row">
                <img src="img/unsplash_OnFW5djcAYc.png" alt="Notebook" id="notebook">
            </div>
        </div>
    </section>
    <section class="data">
        <div class="container-fluid cont-pht">
            <div class="row d-flex justify-content-center align-items-center">
                <div class="col-sm-8 notes">
                    MarkIt allows you to always keep your notes handy. Access them from anywhere
using our secure cloud!
                </div>
                <img src="img/photo1.png" alt="photo1" class="img-fluid col-sm-4 mx-auto girl"
width=">

            </div>
        </div>
    </section>
    <section class=" data">
                <div class="container-fluid cont-pht">
                    <div class="row d-flex justify-content-center align-items-center">
                        <img src="img/photo2.png" alt="photo1" class="img-fluid col-sm-4 mx-auto
girl">
                        <div class="col-sm-8 notes">
                            Make better notes using our Markdown editor.
                        </div>
                    </div>
                </div>
    </section>
</body>

</html>
```

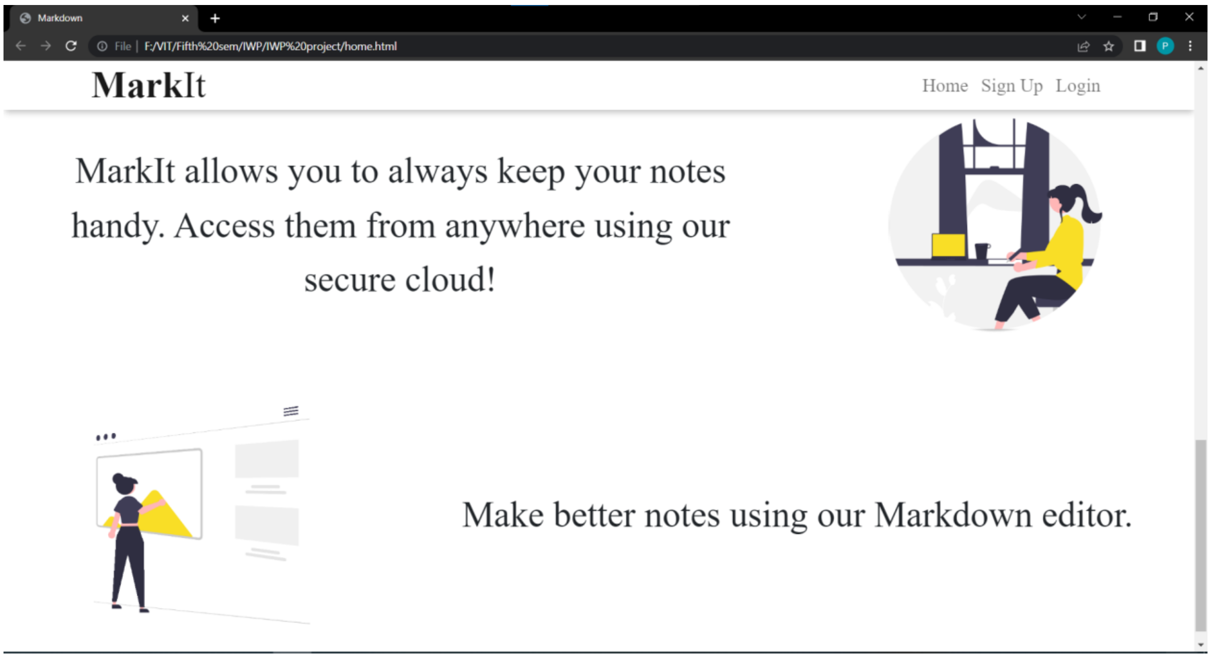
Screenshots:-



MarkIt

Keep your notes handy!





MarkIt allows you to always keep your notes handy. Access them from anywhere using our secure cloud!

Make better notes using our Markdown editor.

Signup page:

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0/dist/css/bootstrap.min.css" rel="stylesheet"
        integrity="sha384-
gH2yIJqKdNHPEq0n4Mqa/HGKIhSkIHeL5AyhkYV8i59U5AR6csBvApHHNl/vI1Bx"
crossorigin="anonymous">
    <!-- <link rel="stylesheet" href="css/bootstrap.min.css"> -->
    <link rel="stylesheet" href="signup.css">
    <title>Markdown</title>
</head>

<body>


    <nav class="navbar navbar-light navbar-expand-md bg-light justify-content-center">
        <div class="container">
            <a href="notePage.html" class="navbar-brand d-flex me-auto">Mark<span>It</span></a>
            <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#collapsingNavbar3">
                <span class="navbar-toggler-icon"></span>
            </button>
            <div class="navbar-collapse collapse w-100" id="collapsingNavbar3">
        <ul class="nav navbar-nav ms-auto w-100 justify-content-end">
          <li class="nav-item">
            <a class="nav-link" href="home.html">Home</a>
          </li>
          <li class="nav-item">
            <a class="nav-link" href="signup.html">Sign Up</a>
          </li>
          <li class="nav-item">
            <a class="nav-link" href="login.html">Login</a>
          </li>
        </ul>
      </div>
    </div>
  </nav>


  <div class="container mt-5">
    <div class="row d-flex justify-content-center">
      <div class="col-md-6">
        <div class="card px-5 py-5" id="form1">
          <div class="form-data" v-if="!submitted">
            <form>
              <!-- Email input -->
              <div class="form-outline mb-4">
                <label class="form-label" for="form2Example1">Email address</label>
                <input type="email" id="form2Example1" class="form-control" />
              </div>

              <!-- Password input -->
              <div class="form-outline mb-4 ">
                <label class="form-label " for="form2Example2">Password</label>
                <input type="password" id="form2Example2" class="form-control" />
              </div>

              <!-- Confirm Password input -->
              <div class="form-outline mb-4 ">
                <label class="form-label " for="form2Example2">Confirm Password</label>
                <input type="password" id="form2Example2" class="form-control" />
              </div>

              <!-- 2 column grid layout for inline styling -->
              <div class="row mb-4 text-center">
                <div class="col d-flex justify-content-center">
                  <!-- Checkbox -->
                  <div class="form-check">
                    <input class="form-check-input" type="checkbox" value=""
id="form2Example31"
                      checked />
```

```
                                        <label class="form-check-label" for="form2Example31"> Remember me
</label>
                                    </div>
                                </div>

                                <div class="col">
                                    <!-- Simple link -->
                                    <a href="#!">Forgot password?</a>
                                </div>
                            </div>


                            <!-- Submit button -->
                            <div class="text-center">
                                <button type="button" class="btn btn-primary btn-block mb-4"
                                    id="login-btn">Login</button>
                            </div>
                            <!-- Register buttons -->
                            <div class="text-center">
                                <p>Not a member? <a href="#!">Sign Up</a></p>
                            </div>
                        </form>
                    </div>
                </div>
            </div>
        </div>
    </div>



</body>

</html>
```

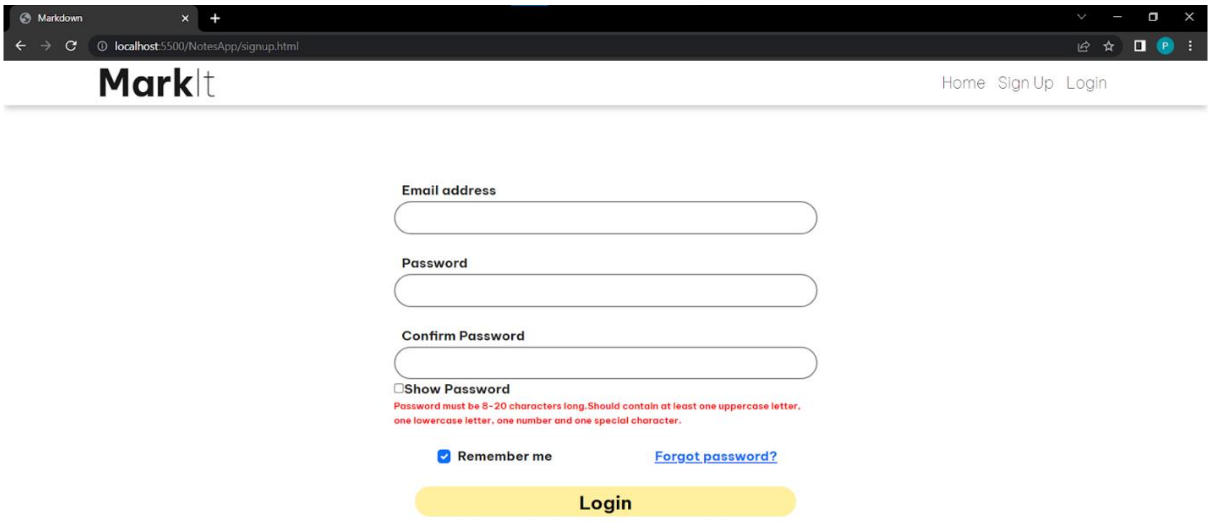
Signup CSS:

```
html,
body {
    width: 100%;
    height: 100%;
    font-family: Be Vietnam Pro;
}


.navbar {
    background: #FFFFFF !important;
    font-size: 24px;
    font-weight: 500;
```

```css
    line-height: 30px;
    box-shadow: 0px 5px 8px -2px #00000040;
}

.navbar-brand {
    font-size: 48px;
    font-weight: 700;
}

.navbar-brand>span {
    font-weight: 100;
}


.card {
    color: #2E2E2E;
    border: none;
    font-family: 'Inter';
    font-style: normal;
    font-weight: 700;
    font-size: 17px;
}

.form-label {
    padding-left: 10px;
    margin-bottom: 1px;
}

#form2Example1,
#form2Example2 {
    width: 100%;
    height: 42px;
    border: 2px solid rgba(0, 0, 0, 0.49);
    border-radius: 42px;
}

#login-btn {
    width: 90%;
    height: 38px;

    background: #FFF0A0;
    border-radius: 65px;
    border: none;

    font-weight: 700;
    font-size: 24px;
    line-height: 29px;
    color: #000000;
    text-align: center;
```

```
}
```

Screenshots:-


Markdown
localhost:5500/NotesApp/signup.html

Email address

Password

Confirm Password

☐ Show Password
Password must be 8-20 characters long.Should contain at least one uppercase letter, one lowercase letter, one number and one special character.

☑ Remember me
Forgot password?

Login page:

```
<!DOCTYPE html>
<html lang="en">

<head>
   <meta charset="utf-8">
   <meta name="viewport" content="width=device-width, initial-scale=1">

   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0/dist/css/bootstrap.min.css"
rel="stylesheet"
       integrity="sha384-
gH2yIJqKdNHPEq0n4Mqa/HGKIhSkIHeL5AyhkYV8i59U5AR6csBvApHHNl/vI1Bx"
crossorigin="anonymous">

   <link rel="preconnect" href="https://fonts.googleapis.com">
   <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
   <link
href="https://fonts.googleapis.com/css2?family=Be+Vietnam+Pro:wght@100;700&family=Lora
&display=swap"
       rel="stylesheet">
```

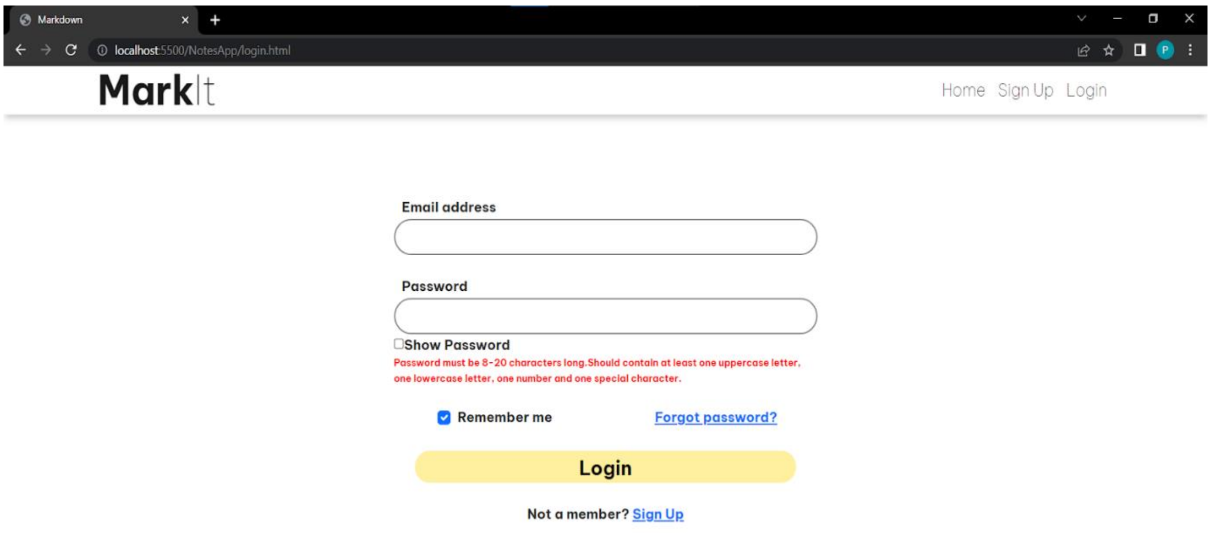
```html
    <link rel="stylesheet" href="css/login.css">
    <title>Markdown</title>
</head>

<body>


    <nav class="navbar navbar-light navbar-expand-md bg-light justify-content-center">
        <div class="container">
            <a href="/" class="navbar-brand d-flex me-auto">Mark<span>It</span></a>
            <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#collapsingNavbar3">
                <span class="navbar-toggler-icon"></span>
            </button>
            <div class="navbar-collapse collapse w-100" id="collapsingNavbar3">
                <ul class="nav navbar-nav ms-auto w-100 justify-content-end">
                    <li class="nav-item">
                        <a class="nav-link" href="/">Home</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="/signup">Sign Up</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="/login">Login</a>
                    </li>
                </ul>
            </div>
        </div>
    </nav>


    <div class="container mt-5">
        <div class="row d-flex justify-content-center">
            <div class="col-md-6">
                <div class="card px-5 py-5" id="form1">
                    <div class="form-data" v-if="!submitted">

                        <!-- <form onsubmit="return false;" id="form"> -->
                        <form action="/login" method="post" id="form">
                            <div class="form-outline mb-4">
                                <label class="form-label" for="email">Email address</label>
                                <input type="email" id="email" name="email" class="form-control" />
                            </div>


                            <div class="form-outline mb-4 ">
                                <label class="form-label " for="password">Password</label>
                        <input type="password" name="password" id="password" class="form-
control" />
                     </div>




                     <div class="text-center">
                        <button type="submit" class="btn btn-primary btn-block mb-4"
                           id="login-btn">Login</button>
                     </div>

                     <div class="text-center">
                        <p>Not a member? <a href="/signup">Sign Up</a></p>
                     </div>
                  </form>
                  <div class="alert alert-danger" role="alert" id="error" style="display :
none;"></div>
                  <div class="alert alert-success" role="alert" id="success" style="display : none;
;"></div>
               </div>
            </div>
         </div>
      </div>
   </div>
   <%- include('layouts/flash-messages') %>
</body>

</html>
```

Login CSS:

```
html,
body {
   width: 100%;
   height: 100%;
   font-family: Be Vietnam Pro;
}


.navbar {
   background: #FFFFFF !important;
   font-size: 24px;
   font-weight: 500;
   line-height: 30px;
   box-shadow: 0px 5px 8px -2px #00000040;
```

```css
}

.navbar-brand {
    font-size: 48px;
    font-weight: 700;
}

.navbar-brand>span {
    font-weight: 100;
}


.card {
    color: #2E2E2E;
    border: none;
    font-family: 'Inter';
    font-style: normal;
    font-weight: 700;
    font-size: 17px;
}

.form-label {
    padding-left: 10px;
    margin-bottom: 1px;
}

#form2Example1,
#form2Example2 {
    width: 100%;
    height: 42px;
    border: 2px solid rgba(0, 0, 0, 0.49);
    border-radius: 42px;
}

#login-btn {
    width: 90%;
    height: 38px;

    background: #FFF0A0;
    border-radius: 65px;
    border: none;

    font-weight: 700;
    font-size: 24px;
    line-height: 29px;
    color: #000000;
    text-align: center;
}
```

Screenshots:-


Markdown
localhost:5500/NotesApp/login.html
MarkIt
Home
Sign Up
Login
Email address
Password
Show Password
Password must be 8-20 characters long.Should contain at least one uppercase letter, one lowercase letter, one number and one special character.
Remember me
Forgot password?
Login
Not a member? Sign Up

Notes Page:

```
<!DOCTYPE html>
<html>

<head>
    <title>Notes</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0/dist/css/bootstrap.min.css"
rel="stylesheet"
        integrity="sha384-
gH2yIJqKdNHPEq0n4Mqa/HGKIhSkIHeL5AyhkYV8i59U5AR6csBvApHHNl/vI1Bx"
crossorigin="anonymous">

    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.1.2/css/all.min.css"
        integrity="sha512-
1sCRPdkRXhBV2PBLUdRb4tMg1w2YPf37qatUFeS7zlBy7jJI8Lf4VHwWfZZfpXtYSLy85pk
m9GaYVYMfw5BC1A=="
        crossorigin="anonymous" referrerpolicy="no-referrer" />
    <script src="https://cdnjs.cloudflare.com/ajax/libs/marked/4.0.18/marked.min.js"
        integrity="sha512-
6MhVG3FdJSogKdAFmdGWvJ/QCBTPhs0dd3ad89H+ZVGJ721bOo2CTlaiG4Ov6l422KGQSt
YrfOD4HASBvO6sZw=="
```

```
        crossorigin="anonymous" referrerpolicy="no-referrer"></script>
    <link rel="preconnect" href="https://fonts.googleapis.com">
    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
    <link
href="https://fonts.googleapis.com/css2?family=Be+Vietnam+Pro:wght@100;700&family=Lora
&display=swap"
        rel="stylesheet">

    <link rel="stylesheet" href="css/notePage.css">
    <script defer src="js/notePage.js"></script>
</head>

<body>
    <nav class="navbar navbar-light navbar-expand-md bg-light justify-content-center">
        <div class="container">
            <a href="/" class="navbar-brand d-flex me-auto">Mark<span>It</span></a>
            <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#collapsingNavbar3">
                <span class="navbar-toggler-icon"></span>
            </button>
            <div class="navbar-collapse collapse w-100" id="collapsingNavbar3">
                <ul class="nav navbar-nav ms-auto w-100 justify-content-end">
                    <li class="nav-item">
                        <a class="nav-link" href="/">Home</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="/signup">Sign Up</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="/">Logout</a>
                    </li>
                </ul>
            </div>
        </div>
    </nav>

    <div class="container mt-5">
        <div class="row d-flex ">
            <div class="col-sm-11 d-flex align-items-center justify-content-center">
                <h1>My Notes: <span id="usern"><b>
                            <% data.forEach(function(d,index){ %>
                                <%- d.username %>
                                    <% }) %>
                    </span>
                    </b>
                </h1>
            </div>
            <!-- <div class="col-sm-1 d-flex justify-content-end align-items-center">
                <button class="add" id="add"><i class="fa-solid fa-plus"></i></button>
            </div> -->
        </div>
    </div>
    <div class="container mt-4">
        <div class="notes">
            <div class="tools d-flex">
                <div class="note-tab d-flex justify-content-front col-sm-10"></div>
                <div class="edit-buttons d-flex justify-content-end col-sm-2">
                    <button class="share" onclick="share()"><i class="fa-solid fa-share"></i></button>
                    <button class=" edit" id="edit-btn" onclick="rendr()"><i class="fa-solid fa-
marker"></i></button>
                    <!-- <button class="delete" id="del-btn"><i class="fa-solid fa-trash-
can"></i></button> -->
                </div>
            </div>
            <div class="main hidden"></div>
            <textarea name="text" id="text" cols="40" rows="10" value=""></textarea>
            <div id="content"></div>
            <form action="/update" method="post" id="form">

                <% data.forEach(function(d,index){ %>
                    <input type="hidden" name="email" value="<%- d.email %>">
                    <input type="hidden" name="username" value="<%- d.username %>">
                    <input type="hidden" id="note1" name="note1" value="<%- d.note1 %>">
                    <input type="hidden" id="note2" name="note2" value="<%- d.note2 %>">
                    <input type="hidden" id="note3" name="note3" value="<%- d.note3 %>">
                    <input type="hidden" id="note4" name="note4" value="<%- d.note4 %>">
                    <% }) %>

                        <div class=" d-flex justify-content-end">
                            <button id="save" type="submit">Save</button>
                        </div>
            </form>
        </div>

    </div>
</body>

</html>
```

<u>**NotePage CSS:**</u>

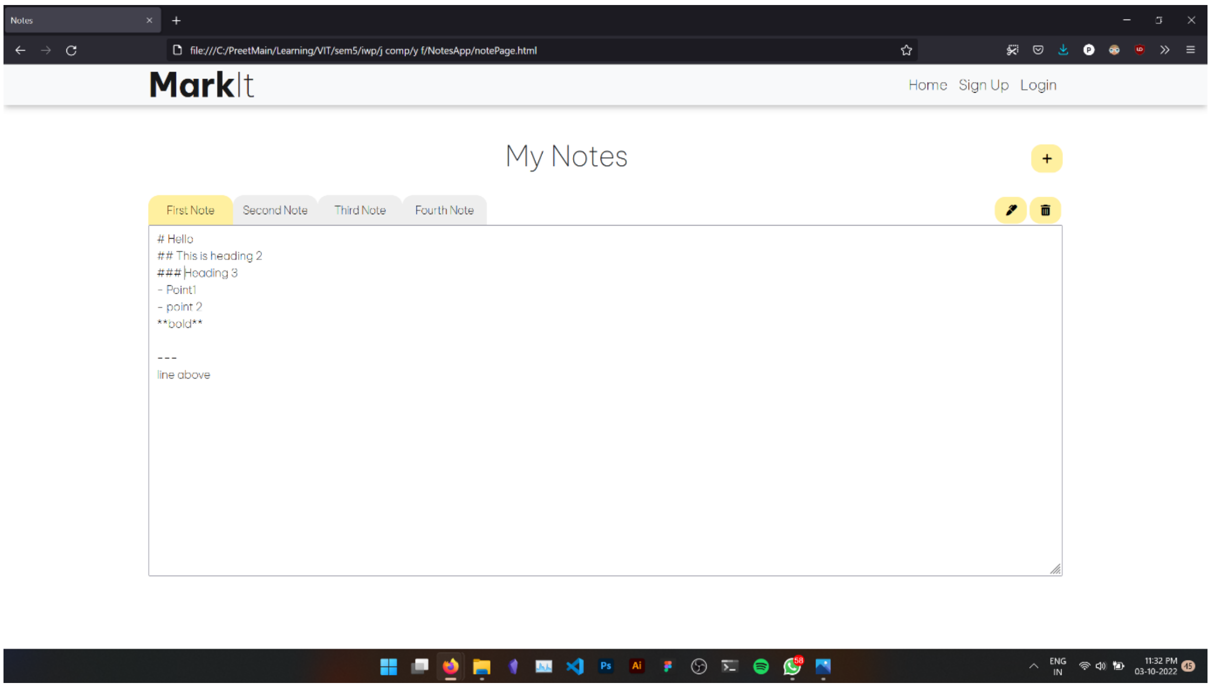
```css
* {
    font-family: 'Be Vietnam Pro';
    /* font-family: 'Lora' !important; */
    margin: 0;
    padding: 0;
}

html,
body {
    width: 100%;
    height: 100%;
}


.nav-item .nav-link {
    color: black;
}

.navbar {
    background: #FFFFFF !important;
    font-size: 20px;
    font-weight: 500;
    line-height: 25px;
    box-shadow: 0px 5px 8px -2px #00000040;
}

.navbar-brand {
    font-size: 48px !important;
    font-weight: 700;
}

.navbar-brand>span {
    font-weight: 100;
}


/* Notepad */
.add {
    background-color: #FFF0A0;
    border: none;
    color: black;
    padding: 0.5rem 1rem;
    cursor: pointer;
    width: 45px;
    border-radius: 15px;
}
```

```css
.note-tab button {
   background-color: #FFF0A0;
   border: none;
   font-size: 15px;
   cursor: pointer;
   width: 120px;
   border-radius: 15px 15px 0 0;
}

.edit-buttons button {
   background-color: #FFF0A0;
   border: none;
   margin: 2px;
   padding: 0.5rem 1rem;
   font-size: 15px;
   cursor: pointer;
   width: 45px;
   border-radius: 15px;
}

.note-tab .tab-s {
   background-color: #eeeeee;
}

.notes textarea {
   outline: none;
   font-family: inherit;
   height: 60vh;
   width: 100%;
}

#content{
   background-color: rgb(255, 255, 255);
   color: #282828;
   display: none;
   padding: 5px 10px 5px;
   border: solid 2px #fff0a0;
   height: 32em;
   overflow: scroll;
}

#text-area{
   padding: 5px 10px 5px;
}
```

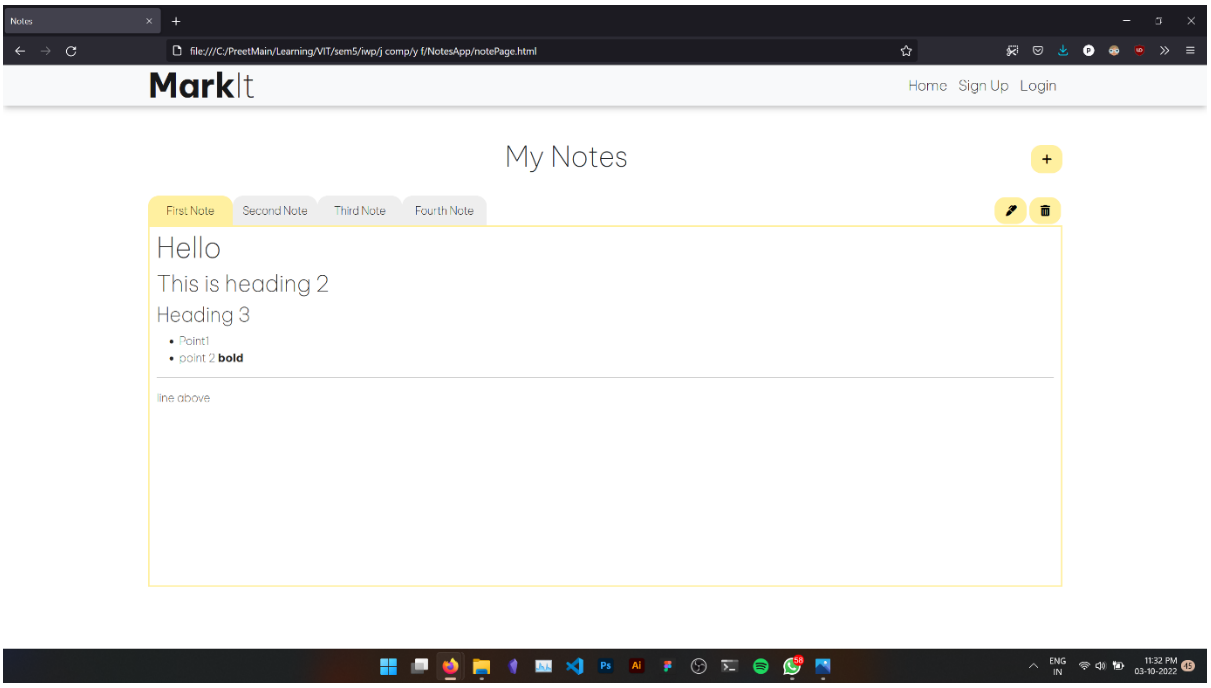
JS:

```js

const notesState = [
    {
        noteName: "First Note",
        content: "Hello",
        id: "note1",
    },
    {
        noteName: "Second Note",
        content: "# Better hello",
        id: "note2",
    },
    {
        noteName: "Third Note",
        content: "Even better hello",
        id: "note3",
    },
    {
        noteName: "Fourth Note",
        content: "Best Hello",
        id: "note4",
    },
];
notesState[0].content = document.getElementById("note1").value;
notesState[1].content = document.getElementById("note2").value;
notesState[2].content = document.getElementById("note3").value;
notesState[3].content = document.getElementById("note4").value;

const inactiveTabName = "tab-s";
const activeTabName = "tab";

const renderActiveTab = (id) => {
    console.log("rendering active tab", id);
    const noteTabsContainer = document.querySelector(".note-tab");
    noteTabsContainer.innerHTML = "";
    notesState.forEach((noteData) => {
        const noteTab = document.createElement("button");
        if (noteData.id === id) {
            noteTab.classList.add(activeTabName);
        } else {
            noteTab.classList.add(inactiveTabName);
        }
        noteTab.innerText = noteData.noteName;
        noteTab.dataset.tabId = noteData.id;
        noteTabsContainer.appendChild(noteTab);
    });
    const noteTabs = document.querySelectorAll(".tab-s");
    noteTabs.forEach((noteTab) => {
        noteTab.addEventListener("click", (e) => {
            renderActiveTab(e.target.dataset?.tabId);
        });
    });

    const activeIndex = notesState.findIndex((noteData) => noteData.id === id);
    const textInput = document.querySelector("textarea");
    textInput.value = notesState[activeIndex].content;
    textInput.oninput = (e) => {
        notesState[activeIndex].content = e.target.value;
        document.getElementById(notesState[activeIndex].id).value = e.target.value;

        console.log(e.target.value);
        // console.log(notesState[activeIndex].id);
        // console.log(activeIndex, e.target.value, notesState[activeIndex]);
    };
};

renderActiveTab(notesState[0].id);
var op = 0;


function rendr() {

    if (op == 0) {
        document.getElementById('text').style.display = "none";
        document.getElementById('content').style.display = "block";
        document.getElementById('save').style.display = "none";
        document.getElementById('content').innerHTML =
marked.parse(document.getElementById('text').value);
        // document.getElementById('content').innerHTML = "test"
        op = 1;
    } else {
        document.getElementById('text').style.display = "block";
        document.getElementById('content').style.display = "none";
        document.getElementById('save').style.display = "block";
        op = 0;
    }
}

function share() {
    document.getElementById('content').innerHTML =
marked.parse(document.getElementById('text').value);
    var divContents = document.getElementById("content").innerHTML;
    var a = window.open('', '', 'height=500, width=500');
    a.document.write('<html>');
    a.document.write('<body >');
    a.document.write(divContents);
    a.document.write('</body></html>');
    a.document.close();
    a.print();
}
```

<u>Screenshots:-</u>



```
# Hello
## This is heading 2
### Heading 3
- Point1
- point 2
**bold**

---
line above
```



Hello

This is heading 2

Heading 3

- Point1
- point 2 **bold**

---

line above

NodeJS:-

Pages.js:

The pages.js is used to render the webpage whenever the user wants to access it. When a user tries to go on any other part of the website, the browser send a request to pages.js and then the webpage is rendered and the necessary actions are taken.

```
const express = require("express");
const router = express.Router();
const db = require("../routes/db-config");
const bcrypt = require("bcryptjs");
const { body, validationResult } = require('express-validator');

router.get("/", async (req, res, next) => {
    res.render("index");
});



router.get("/signup", (req, res, next) => {

    res.render("signup");
});

router.get("/login", async (req, res, next) => {
    res.render("login");
});

router.get("/notePage", async (req, res, next) => {
    res.render("notePage");
});

router.post('/login', async (req, res, next) => {
    const { email, password: Npassword } = req.body;
    if (!email || !Npassword) {
        req.flash('error', "Please Enter your email and password");
        res.render('login', { messages: req.flash() });
        return;
    }
    else {
        db.query('SELECT * FROM users WHERE email= ?', [email], async (err, result) => {
            if (err) throw err;
            if (!result.length || !await bcrypt.compare(Npassword, result[0].password)) {
                req.flash('error', "Incorrect Email or password");
                res.render('login', { messages: req.flash() });
                return;
            }
            else {
                // console.log(result[0].username)
                const username = result[0].username;
                db.query('SELECT * from notes where username= ?', [username], (err, results) => {
                    if (err) throw err
                    else {
                        res.render("notepage", { data: results });
                        // console.log(results);
                    }

                });
            }
        });
    }
});

router.post('/signup', [
    body('email').trim().isEmail().withMessage('Email must be a valid
email').normalizeEmail().toLowerCase(),
    body('password').trim().isLength(2).withMessage('Username length too short, min 5 char
required'),
    body('password').trim().isLength(2).withMessage('Password length short, min 2 char
required'),
    body('password2').custom((value, { req }) => {
        if (value !== req.body.password) {
            throw new Error('Password do not match');
        }
        return true;
    })
], async (req, res, next) => {
    const errors = validationResult(req);
    if (!errors.isEmpty()) {
        errors.array().forEach(error => {
            req.flash('error', error.msg)
        });
        res.render('signup', { messages: req.flash() });
        return;
    }

    const { email, username: username, password: Npassword } = req.body;
    if (!email || !Npassword) {
        res.flash('error', "Please Enter your email and password");
        res.render('signup', { messages: req.flash() });
        return;
    }
    else {
        db.query('SELECT email FROM users WHERE email= ?', [email], async (err, result) => {
            if (err) throw err;
            if (result[0]) {
                req.flash('error', "Email has already been registered");
                res.render('signup', { messages: req.flash() });
                return;
            }
            else {
                const password = await bcrypt.hash(Npassword, 8);
                db.query('INSERT INTO users SET ?', { email: email, username: username, password:
password }, (error, results) => {
                    if (error) throw error;
                    req.flash('success', `Registered succesfully, you can now login`);
                    db.query('INSERT INTO notes SET ?', { email: email, username: username, note1:
"Hello", note2: "# Better hello", note3: "Even better hello", note4: "Best Hello" }, (error, results)
=> {
                        if (error) throw error;
                    });



                    res.render('signup', { messages: req.flash() });
                    return;
                    // res.send({ status: "success", success: "User has been registered" });
                });
            }
        });
    }
});

router.post("/update", async (req, res, next) => {
    const { text, username, email, note1, note2, note3, note4 } = req.body;
    // console.log(username);
    db.query('UPDATE notes SET ? where username = ?', [{ note1: note1, note2: note2, note3:
note3, note4: note4, }, username], (error, result) => {
        if (error) throw error;
    });
    var ret = [];
    db.query('SELECT * from notes where username= ?', [username], (err, results) => {
        if (err) throw err
        else {
            Object.keys(results).forEach(function (key) {
                ret = results[key];
            });
            // console.log("HI" + ret.email);
            db.query('UPDATE notes SET ? where username = ?', [{ note1: ret.note1, note2:
ret.note2, note3: ret.note3, note4: ret.note4, }, username], (error, result) => {
                if (error) throw error;
            });
            res.render("notepage", { data: results });
            // console.log(results);
        }
    });

    // res.render('notePage', { username });
});



module.exports = router;
```

Login:

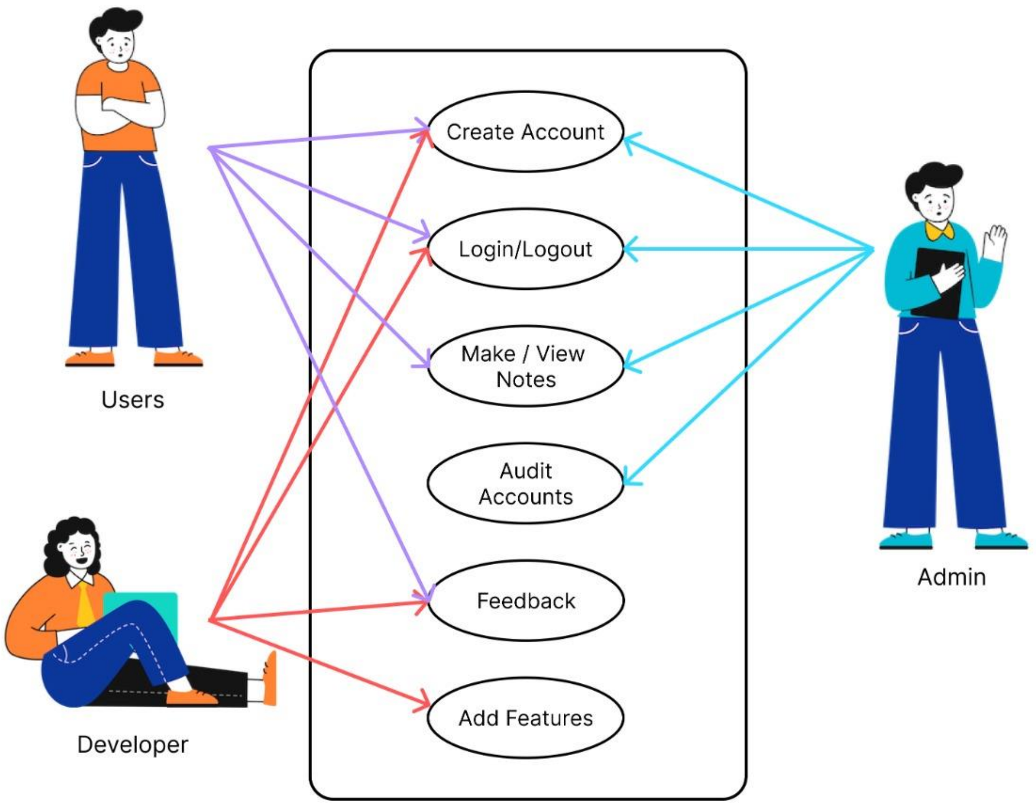
```
const db = require("../routes/db-config");
const bcrypt = require("bcryptjs");
// const jwt = require("jsonwebtoken");

const login = async (req, res) => {
    const { email, password: Npassword } = req.body;
    if (!email || !Npassword)
        return res.json({ status: "error", error: "Please Enter your email and password" });
    else {
        db.query('SELECT * FROM users WHERE email= ?', [email], async (err, result) => {
            if (err) throw err;
            if (!result.length || !await bcrypt.compare(Npassword, result[0].password))
                return res.json({ status: "error", error: "Incorrect Email or password" });
            else {
                return res.json({ status: "success", success: "User has been logged In" });
            }
        });
    }
}
module.exports = login;
```

db-config.js:

Connecting to the database.

```
const sql = require("mysql");


let db = sql.createConnection({
    host: 'localhost',
    user: 'root',
    password: 'password',
    database: 'user_reg'
});

module.exports = db;
```

Use Case Diagram:

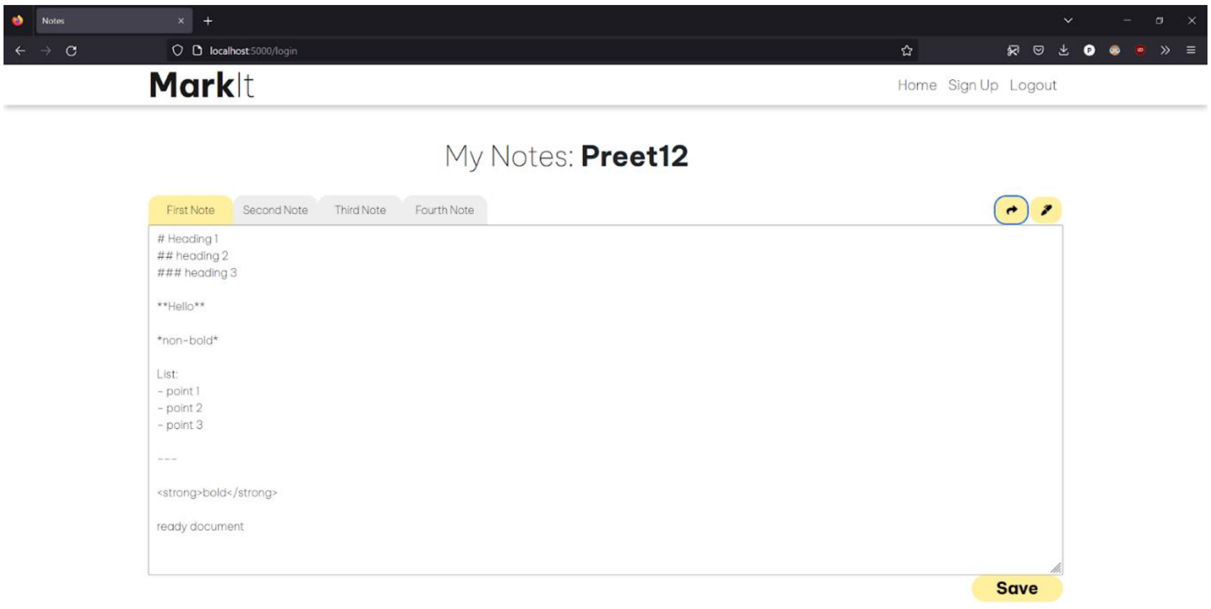


Results

(Database design, Table schema and screen shots with explanation)
Eg:

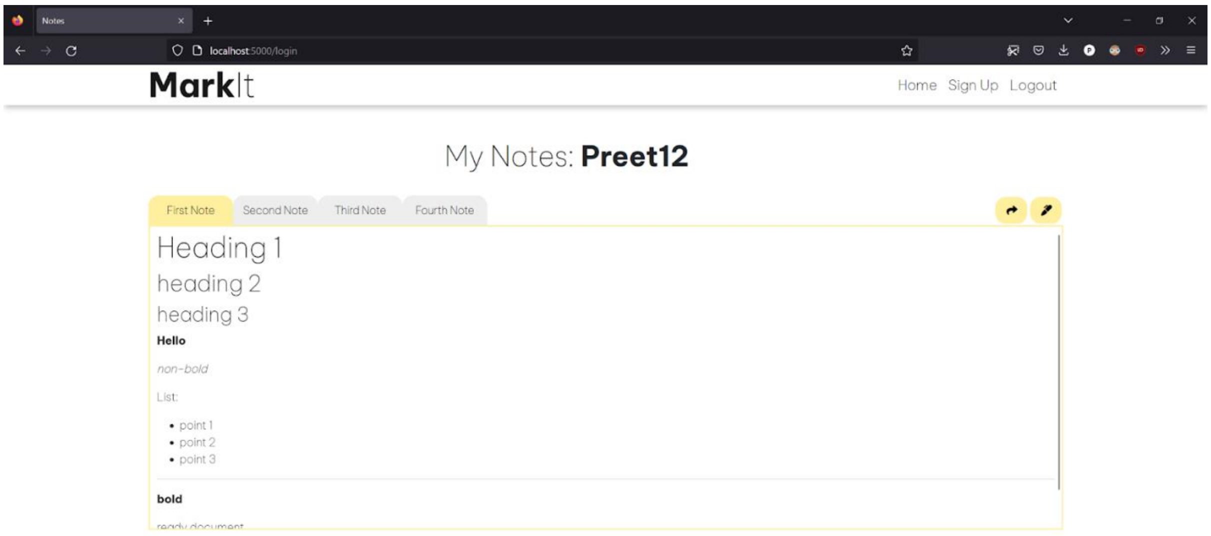
```
# Heading 1
## heading 2
### heading 3


**Hello**


*non-bold*


List:
- point 1
- point 2
- point 3
---
<strong>bold</strong>


ready document
```

Screenshot of plain text:



```
# Heading 1
## heading 2
### heading 3

**Hello**

*non-bold*

List:
- point 1
- point 2
- point 3

---

<strong>bold</strong>

ready document
```

Screenshot of Markdown view:


Notes

localhost:5000/login

MarkIt

Home Sign Up Logout

My Notes: **Preet12**

First Note | Second Note | Third Note | Fourth Note

Heading 1

heading 2

heading 3

Hello

non-bold

List:

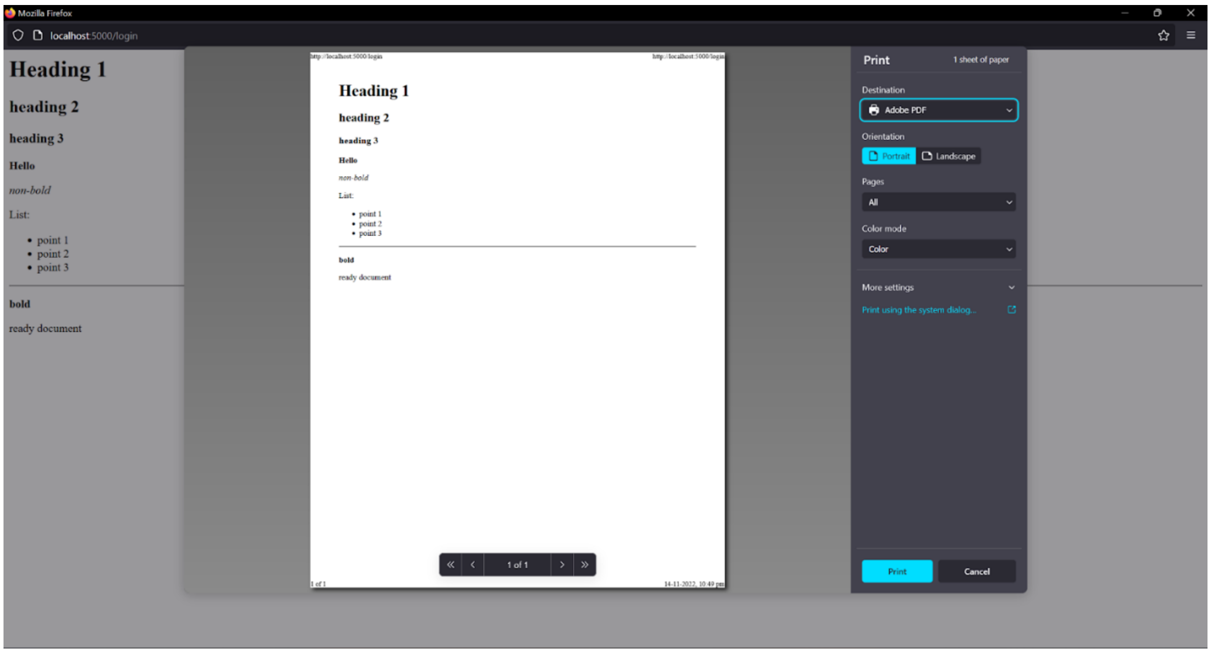
- point 1
- point 2
- point 3

---

bold

ready document

Screenshot of the shared doc:


Mozilla Firefox

localhost:5000/login

Heading 1

heading 2

heading 3

Hello

non-bold

List:

- point 1
- point 2
- point 3

---

bold

ready document

http://localhost:5000/login

1 of 1

14-11-2022, 10:49 pm

Print 1 sheet of paper

Destination: Adobe PDF

Orientation: [✓ Portrait] Landscape

Pages: All

Color mode: Color

More settings

Print using the system dialog...

Print Cancel

Conclusion:

Notes made in the physical class gives the perfect idea of the topic such as headings, important points,highlighted lines,keywords,so on.MarkIt is the editor which allows user to edit the notes while making them and makes it fluent and much simpler to edit the notes.It keeps the user credentials secure by using hash algorithm on the password.The hashed password is saved in the database which does not reveal the original password.It allows users to format the note just by adding the appropriate elements.We can call them as shortcuts to change a text to a specific format.Just one click is enough to format and organize the note.The notes made in the notes page are saved in the database and are easily accessible to user.These notes can also be shared by converting them to pdf and saving it in the device.MarkIt is the reliable website to make notes and access them anywhere user wants.

Code:

https://github.com/Harsh-Wardhan-1/Mark-It

Video:

https://drive.google.com/file/d/1W0t54rcZkCrUQz4GgdaDIFsV2KnSP700/view?usp=sharing