A Micro Project report on

MY NOTE – A HTML5 APP

Submitted to the CMR Institute of Technology in partial fulfillment of the requirement of the

FULL STACK WEB DEVELOPMENT LAB

of

III-B.Tech. II-Semester

in

Department of Computer Science and Engineering

Submitted by

A. Anil	**20R01A05J3**
B. Shradha	**20R01A05J8**
K. Sai Raj Nikhil	**20R01A05L3**
J. Aravind	**20R01A05L4**
K. Sreeja	**20R01A05M0**
K. Abhisaik	**20R01A05M1**

Under the Guidance Of

Mrs. A. Mahalakshmi

(Assistant professor, CSE Dept)

CMR INSTITUTE OF TECHNOLOGY

(UGC AUTONOMUS)
(Approved by AICTE,Affiliated to JNTU,Kukatpally,Hyderabad)
Kandlakoya,Medchal Road,Hyderabad

2022-2023

CMR INSTITUTE OF TECHNOLOGY

(UGC AUTONOMUS)

(Approved by AICTE, Affiliated to JNTU, Kukatpally, Hyderabad)

Kandlakoya, Medchal Road, Hyderabad.

Department of Computer Science and Engineering

CERTIFICATE

This is to certify that a **Micro Project** entitled with My Note – A HTML5 App **,** is being submitted by

A. Anil	**20R01A05J3**
B. Shradha	**20R01A05J8**
K. Sai Raj Nikhil	**20R01A05L3**
J. Aravind	**20R01A05L4**
K. Sreeja	**20R01A05M0**
K. Abhisaik	**20R01A05M1**

in partial fulfillment of the requirement for completion of the full stack web technologies lab of III- B.Tech. II- Semester .

Signature of Faculty **Signature of Course Co-ordinator** **Signature of HOD**

ACKNOWLEDGEMENT

I am extremely grateful to **Dr. M. Janga Reddy**, **Director**, **Dr. B. Satyanarayana**, **Principal** and **Mr. A. Prakash**, **Head of Department**, **Department of Computer Science and Engineering**, CMR Institute of Technology for their inspiration and valuable guidance during entire duration.

I am extremely thankful to **Mrs. A. Mahalakshmi**, **Assistant professor**, Computer Science and Engineering department, CMR Institute of Technology for her constant guidance, encouragement and moral support throughout the project.

A. Anil	**20R01A05J3**
B. Shradha	**20R01A05J8**
K. Sai Raj Nikhil	**20R01A05L3**
J. Aravind	**20R01A05L4**
K. Sreeja	**20R01A05M0**
K. Abhisaik	**20R01A05M1**

INDEX

ABSTRACT

This project is about how the user can make a note in an app named MyNote which is developed with HTML5. Advantages of note-making are it requires active listening, helps you learn, helps you remember information, makes studying easier, is personalized, teaches you to summarize. Note making is not just about writing down everything you hear or read. It is a process of reviewing, connecting and synthesizing ideas from your lectures or reading. Notes is a good helper to manage your schedules and notes. It gives you a quick and simple notepad editing experience when you write notes, memo, email, message, shopping list and to do list. It makes to take a note easier than any other notepad and memo apps. Note taking apps make it simple to keep up with your information. Beyond the convenience in your pocket, most note taking tools also provide remarkable accessibility. You can add, copy and edit your notes from a laptop, tablet or phone. This helps you transfer notes to work accounts or easily share them with friends. This project makes use of front-end technologies such as HTML, CSS and JAVASCRIPT. Firstly, the user has to login to the page using his/her credentials. A page will be opened with an option to add a note. User can add any note by click on the icon shown on the page displayed. Now a pop up window is displayed on the screen to set the title and description for your note. After, entering the blanks under the title and description, click on add note button to set the title and description to make your note. Then the note you created will be saved and displayed on the screen through which you can access for future usage. Not only a single note, the user can create multiple notes with the same process as said above. The user can edit the created note when there are any changes to be made and when the note is never required you can delete it. The app is designed to make note-taking convenient and accessible for the user. Additionally, the notes can be accessed from any device, including a laptop, tablet, or phone, making them easy to transfer or share with others. The app is user-friendly, easy to use, and helps to streamline the note-taking process, making it an excellent tool for anyone who needs to take notes regularly.

1. Introduction

Mobile applications have become an integral part of our lives, and note-taking apps have gained significant popularity among users due to their convenience and accessibility.

Here are some of the key reasons why note-making in mobile applications is important:

- Easy to use: Note-making apps are usually designed to be simple and user-friendly, making it easy to take notes on-the-go. Users can quickly jot down their thoughts or ideas without having to worry about finding a pen and paper.
- Convenience: Note-making apps allow users to access their notes from anywhere, at any time. Whether it's a meeting, a lecture, or a random idea that pops up, note-making apps make it easy to capture information on the spot.
- Organization: Note-making apps provide an efficient way to organize information. Users can categorize their notes, add tags, and search for specific keywords, making it easy to find what they need quickly.
- Collaboration: Many note-making apps allow users to share their notes with others, making it easy to collaborate on projects or share information with colleagues.
- Memory retention: Research has shown that taking notes can improve memory retention and help with learning. Note-making apps make it easy to review and revise notes, which can help with long-term retention of information.

Firstly, it provides convenience and accessibility. With a note-taking app on your mobile device, you can quickly jot down ideas, thoughts, or important information on-the-go. You don't have to carry a separate notebook or pen with you everywhere you go, and you can easily access your notes from any device with internet connectivity. This allows you to be more productive and efficient, and helps you stay organized.

Secondly, note-taking apps can improve learning and memory retention. When you take notes, you are actively engaging with the material and summarizing important information. This helps to reinforce your understanding of the topic and increases the likelihood that you will remember the information later.

Thirdly, mobile note-taking apps offer a range of features that can enhance your note-taking experience. For example, many note-taking apps allow you to attach images, voice recordings, and web links to your notes, making them more detailed and informative. Some apps also offer features such as reminders, to-do lists, and collaboration tools, allowing you to work more effectively and efficiently.

1.1 HTML:

- HTML stands for Hyper Text Markup Language. It is a markup language used to create the structure and content of web pages. HTML provides a set of markup tags or codes that are used to define various elements of a web page, such as headings, paragraphs, links, images, and other media.
- HTML is a foundational technology for creating web pages and is often used in conjunction with other technologies such as CSS (Cascading Style Sheets) and JavaScript to create dynamic and interactive web pages.
- HTML documents are composed of a series of HTML elements, which consist of opening and closing tags with content in between. The content can include text, images, videos, audio, and other multimedia. These elements are used to create the structure of a web page, including headings, paragraphs, lists, tables, forms, and more.
- HTML is a standard markup language, which means that it is recognized and understood by web browsers, making it a universal language for creating web pages.

1.2 CSS:

- CSS stands for Cascading Style Sheets. It is a style sheet language used to define the presentation and styling of HTML documents. CSS is used to separate the presentation of a web page from its content, making it easier to manage and update the style of a web page.
- CSS provides a set of rules that are used to define the style of different elements on a web page, such as the font size, color, background, layout, and more. CSS allows developers to control the visual appearance of a web page, including its layout and design, without changing the content of the web page.
- CSS is often used in conjunction with HTML and JavaScript to create dynamic and interactive web pages. By separating the presentation of a web page from its content, CSS makes it easier to maintain a consistent look and feel across multiple pages of a website.
- CSS uses a cascading style sheet model, which means that multiple style rules can be applied to an element, and the browser will use the most specific rule to apply the style. This allows developers to create complex style rules and apply them to different elements on a web page.

1.3 JavaScript:

- JavaScript is a high-level, object-oriented programming language used primarily to create interactive web pages and web applications. JavaScript is often used in conjunction with HTML and CSS to add dynamic and interactive elements to a web page.

- JavaScript is a client-side scripting language, which means that it runs in the user's web browser rather than on the web server. This allows developers to create dynamic and interactive web pages that respond to user actions and events, without requiring a page refresh or server round-trip.

- JavaScript provides a rich set of features and functionality, including support for variables, data types, operators, control structures, functions, objects, and more. JavaScript also provides a set of built-in objects and functions, such as the Document Object Model (DOM) and the Browser Object Model (BOM), which allow developers to interact with the browser and manipulate the content and behavior of a web page.

- JavaScript is a versatile language that can be used for a variety of purposes, including web development, server-side programming, game development, and more. It is supported by all major web browsers and is a key technology for creating modern and responsive web applications.

2. Source Code

index1.html

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
body {font-family: Arial, Helvetica, sans-serif;}
form {border: 3px solid #f1f1f1;}

input[type=text], input[type=password] {
  width: 100%;
  padding: 12px 20px;
  margin:  8px 0;
  display: inline-block;
  border: 1px solid #ccc;
  box-sizing: border-box;
}

button {
  background-color: #04AA6D;
  color: white;
  padding: 14px 20px;
  margin: 8px 0;
  border: none;
```

```
  cursor: pointer;
  width: 100%;
}

button:hover {
  opacity: 0.8;
}

.cancelbtn {
  width: auto;
  padding: 10px 18px;
  background-color: #f44336;
}

.imgcontainer {
  text-align: center;
  margin: 24px 0 12px 0;
}


.container {
  padding: 16px;
}

span.psw {
  float: right;
  padding-top: 16px;
}

/* Change styles for span and cancel button on extra small screens */
@media screen and (max-width: 300px) {
  span.psw {
     display: block;
     float: none;
  }
  .cancelbtn {
     width: 100%;
  }
}
</style>
</head>
```

```
<body>

<h2>Login Form</h2>

<form name="myForm" action="/action_page.php" method="post">
  <div class="container">
    <label for="uname"><b>Username</b></label>
    <input type="text" placeholder="Enter Username" id="uname" required>

    <label for="psw"><b>Password</b></label>
    <input type="password" placeholder="Enter Password" id="psw" required>

          <button type="submit">Login</button>

    <label>
      <input type="checkbox" checked="checked" name="remember"> Remember
                                                    me
    </label>
  </div>

  <div class="container" style="background-color:#f1f1f1">
    <button type="button" class="cancelbtn">Cancel</button>
    <span class="psw">Forgot <a href="#">password?</a></span>
  </div>
</form>
<script>
document.querySelector('form').addEventListener('submit', function(event) {
  // prevent the default form submission behavior
  event.preventDefault();
  // redirect to the new page
  var un=document.getElementById("uname").value;
  var pw=document.getElementById("psw").value;
  if(un=="student" && pw=="xyz"){
  window.location.href = 'index.html';
  }
  else{
    alert("Invalid");
  }
});
</script>
</body>
```

```
</html>
```

index.html

```
<!DOCTYPE html>
<!-- Coding By CodingNepal - youtube.com/codingnepal -->
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <title>Notes App in JavaScript | CodingNepal</title>
    <link rel="stylesheet" href="style.css">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <!-- Iconscout Link For Icons -->
    <link rel="stylesheet"
                                        href="https://unicons.iconscout.com/releas
                                        e/v4.0.0/css/line.css">
  </head>
  <body>
    <div style="text-align: center; background-color: #575757; color: #fff;">
      <h1 style="padding-left: -40%;">My Note App</h1>
      <p>make a note</p>
    </div>
    <div class="popup-box">
      <div class="popup">
        <div class="content">
          <header>
            <p></p>
            <i class="uil uil-times"></i>
          </header>
          <form action="#">
            <div class="row title">
              <label>Title</label>
              <input type="text" spellcheck="false">
            </div>
            <div class="row description">
              <label>Description</label>
              <textarea spellcheck="false"></textarea>
            </div>
            <button></button>
          </form>
        </div>
      </div>
```

```
    </div>
    <div class="wrapper">
      <li class="add-box">
        <div class="icon"><i class="uil uil-plus"></i></div>
        <p>Add new note</p>
      </li>
    </div>
    <script src="script.js"></script>

  </body>
</html>
```

script.js

```
const addBox = document.querySelector(".add-box"),
popupBox = document.querySelector(".popup-box"),
popupTitle = popupBox.querySelector("header p"),
closeIcon = popupBox.querySelector("header i"),
titleTag = popupBox.querySelector("input"),
descTag = popupBox.querySelector("textarea"),
addBtn = popupBox.querySelector("button");

const months = ["January", "February", "March", "April", "May", "June", "July",
            "August", "September", "October", "November", "December"];
const notes = JSON.parse(localStorage.getItem("notes") || "[]");
let isUpdate = false, updateId;

addBox.addEventListener("click", () => {
    popupTitle.innerText = "Add a new Note";
    addBtn.innerText = "Add Note";
    popupBox.classList.add("show");
    document.querySelector("body").style.overflow = "hidden";
    if(window.innerWidth > 660) titleTag.focus();
});

closeIcon.addEventListener("click", () => {
    isUpdate = false;
    titleTag.value = descTag.value = "";
    popupBox.classList.remove("show");
    document.querySelector("body").style.overflow = "auto";
});
```

```
function showNotes() {
    if(!notes) return;
    document.querySelectorAll(".note").forEach(li => li.remove());
    notes.forEach((note, id) => {
        let filterDesc = note.description.replaceAll("\n", '<br/>');
        let liTag = `<li class="note">
                        <div class="details">
                            <p>${note.title}</p>
                            <hr>
                            <p>${filterDesc}</p>
                        </div>
                        <div class="bottom-content">
                            <p>${note.date}</p>
                            <div class="settings">
                                <i onclick="showMenu(this)" class="uil uil-ellipsis-h"></i>
                                <ul class="menu">
                                    <li style="color: #575757" onclick="updateNote(${id},
                                                '${note.title}', '${filterDesc}')"><i
                                                class="uil uil-pen"></i>Edit</li>
                                    <li style="color: #575757" onclick="deleteNote(${id})"><i
                                                class="uil uil-trash"></i>Delete</li>
                                </ul>
                            </div>
                        </div>
                    </li>`;
        addBox.insertAdjacentHTML("afterend", liTag);
    });
}
showNotes();

function showMenu(elem) {
    elem.parentElement.classList.add("show");
    document.addEventListener("click", e => {
        if(e.target.tagName != "I" || e.target != elem) {
            elem.parentElement.classList.remove("show");
        }
    });
}

function deleteNote(noteId) {
    let confirmDel = confirm("Are you sure you want to delete this note?");
    if(!confirmDel) return;
```

```
    notes.splice(noteId, 1);
    localStorage.setItem("notes", JSON.stringify(notes));
    showNotes();
}

function updateNote(noteId, title, filterDesc) {
    let description = filterDesc.replaceAll('<br/>', '\r\n');
    updateId = noteId;
    isUpdate = true;
    addBox.click();
    titleTag.value = title;
    descTag.value = description;
    popupTitle.innerText = "Update a Note";
    addBtn.innerText = "Update Note";
}

addBtn.addEventListener("click", e => {
    e.preventDefault();
    let title = titleTag.value.trim(),
    description = descTag.value.trim();

    if(title || description) {
        let currentDate = new Date(),
        month = months[currentDate.getMonth()],
        day = currentDate.getDate(),
        year = currentDate.getFullYear();

        let noteInfo = {title, description, date: `${month} ${day}, ${year}`}
        if(!isUpdate) {
            notes.push(noteInfo);
        } else {
            isUpdate = false;
            notes[updateId] = noteInfo;
        }
        localStorage.setItem("notes", JSON.stringify(notes));
        showNotes();
        closeIcon.click();
    }
});
```

style.css

```
//* Import Google Font - Poppins */
@importurl('https://fonts.googleapis.com/css2?family=Poppins:wght@400;500;60
                                                    0;700&display=swap');
*{
  margin: 0;
  padding: 0;
  box-sizing: border-box;
  font-family: 'Poppins', sans-serif;
}
body{
  background: #88ABFF;
}
::selection{
  color: #fff;
  background: #618cf8;
}
.wrapper{
  margin: 50px;
  display: grid;
  gap: 25px;
  grid-template-columns: repeat(auto-fill, 265px);
}
.wrapper li{
  height: 250px;
  list-style: none;
  border-radius: 5px;
  padding: 15px 20px 20px;
  background: #fff;
  box-shadow: 0 4px 8px rgba(0,0,0,0.05);
}
.add-box, .icon, .bottom-content,
.popup, header, .settings .menu li{
  display: flex;
  align-items: center;
  justify-content: space-between;
}
.add-box{
  cursor: pointer;
  flex-direction: column;
```

```
  justify-content: center;
}
.add-box .icon{
  height: 78px;
  width: 78px;
  color: #88ABFF;
  font-size: 40px;
  border-radius: 50%;
  justify-content: center;
  border: 2px dashed #88ABFF;
}
.add-box p{
  color:   #88ABFF;
  font-weight:  500;
  margin-top: 20px;
}
.note{
  display: flex;
  flex-direction: column;
  justify-content: space-between;
}

.note:hover {
  cursor: pointer;
  background-color: #575757;
  transform: scaleX(10px);
  transition: all 0.5s;
  color: #fff;
}
.add-box:hover {
  cursor: pointer;
  background-color: #575757;
  transform: scaleX(10px);
  transition: all 0.5s;
}

.note .details{
  max-height: 165px;
  overflow-y: auto;
}
.note .details::-webkit-scrollbar,
```

```
.popup textarea::-webkit-scrollbar{
  width: 0;
}
.note .details:hover::-webkit-scrollbar,
.popup textarea:hover::-webkit-scrollbar{
  width: 5px;
}
.note .details:hover::-webkit-scrollbar-track,
.popup textarea:hover::-webkit-scrollbar-track{
  background: #f1f1f1;
  border-radius: 25px;
}
.note  .details:hover::-webkit-scrollbar-thumb,
.popup textarea:hover::-webkit-scrollbar-thumb{
  background: #e6e6e6;
  border-radius: 25px;
}
.note p{
  font-size: 22px;
  font-weight: 500;
}
/* .note span{
  display: block;
  color: #575757;
  font-size: 16px;
  margin-top: 5px;
} */
.note  p{
  display: block;
  /* color: #575757; */
  font-size: 16px;
  margin-top: 5px;
}
.note .bottom-content{
  padding-top: 10px;
  border-top: 1px solid #ccc;
}
/* .bottom-content span{
  color: #6D6D6D;
  font-size: 14px;
} */
```

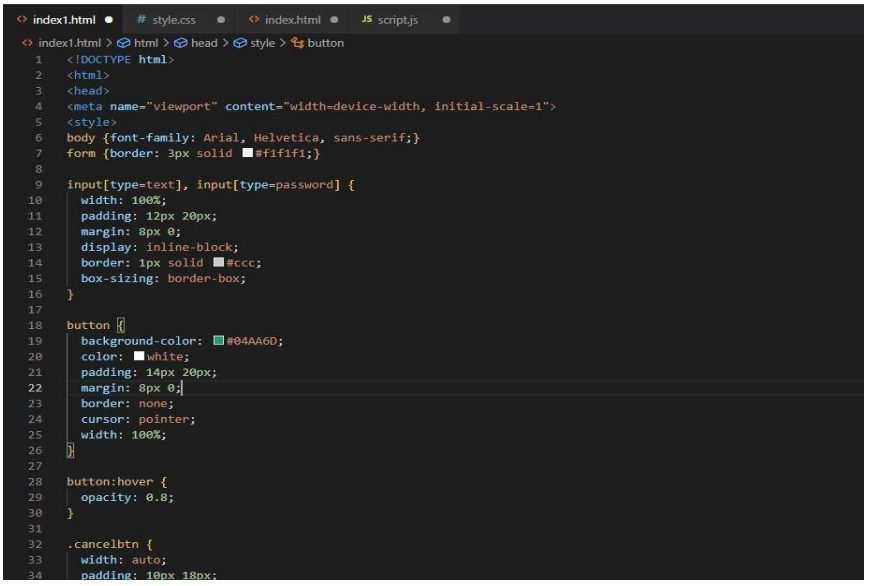
```
.bottom-content p{
  /* color: #6D6D6D; */
  font-size: 14px;
}
.bottom-content .settings{
  position: relative;
}
.bottom-content .settings i{
  color: #6D6D6D;
  cursor: pointer;
  font-size: 15px;
}
.settings .menu{
  z-index: 1;
  bottom: 0;
  right: -5px;
  padding: 5px 0;
  background: #fff;
  position: absolute;
  border-radius: 4px;
  transform: scale(0);
  transform-origin: bottom right;
  box-shadow: 0 0 6px rgba(0,0,0,0.15);
  transition: transform 0.2s ease;
}
.settings.show .menu{
  transform: scale(1);
}
.settings .menu li{
  height: 25px;
  font-size: 16px;
  margin-bottom: 2px;
  padding: 17px 15px;
  cursor: pointer;
  box-shadow: none;
  border-radius: 0;
  justify-content: flex-start;
}
.menu li:last-child{
  margin-bottom: 0;
}
```

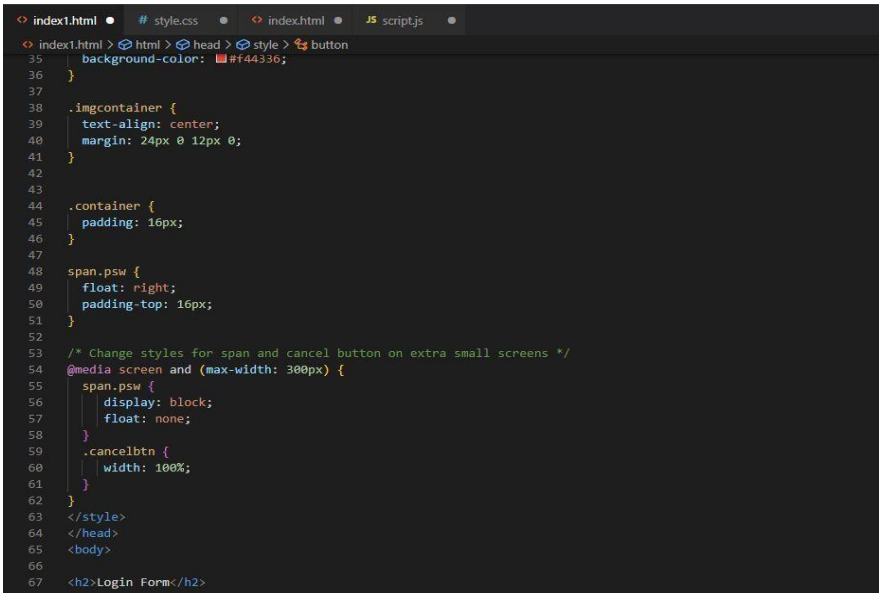
```
.menu li:hover{
  background: #f5f5f5;
}
.menu li i{
  padding-right: 8px;
}
.popup-box{
  position: fixed;
  top: 0;
  left: 0;
  z-index: 2;
  height: 100%;
  width: 100%;
  background: rgba(0,0,0,0.4);
}
.popup-box .popup{
  position:  absolute;
  top: 50%;
  left: 50%;
  z-index: 3;
  width: 100%;
  max-width: 400px;
  justify-content: center;
  transform: translate(-50%, -50%) scale(0.95);
}
.popup-box, .popup{
  opacity: 0;
  pointer-events: none;
  transition: all 0.25s ease;
}
.popup-box.show, .popup-box.show .popup{
  opacity: 1;
  pointer-events: auto;
}
.popup-box.show .popup{
  transform: translate(-50%, -50%) scale(1);
}
.popup .content{
  border-radius: 5px;
  background: #fff;
  width: calc(100% - 15px);
```

```
  box-shadow: 0 0 15px rgba(0,0,0,0.1);
}
.content header{
  padding: 15px 25px;
  border-bottom: 1px solid #ccc;
}
.content header p{
  font-size: 20px;
  font-weight: 500;
}
.content header i{
  color: #8b8989;
  cursor: pointer;
  font-size: 23px;
}
.content form{
  margin: 15px 25px 35px;
}
.content form .row{
  margin-bottom: 20px;
}
form .row label{
  font-size: 18px;
  display: block;
  margin-bottom: 6px;
}
form :where(input, textarea){
  height: 50px;
  width: 100%;
  outline: none;
  font-size: 17px;
  padding: 0 15px;
  border-radius: 4px;
  border: 1px solid #999;
}
form :where(input, textarea):focus{
  box-shadow: 0 2px 4px rgba(0,0,0,0.11);
}
form .row textarea{
  height: 150px;
  resize: none;
```

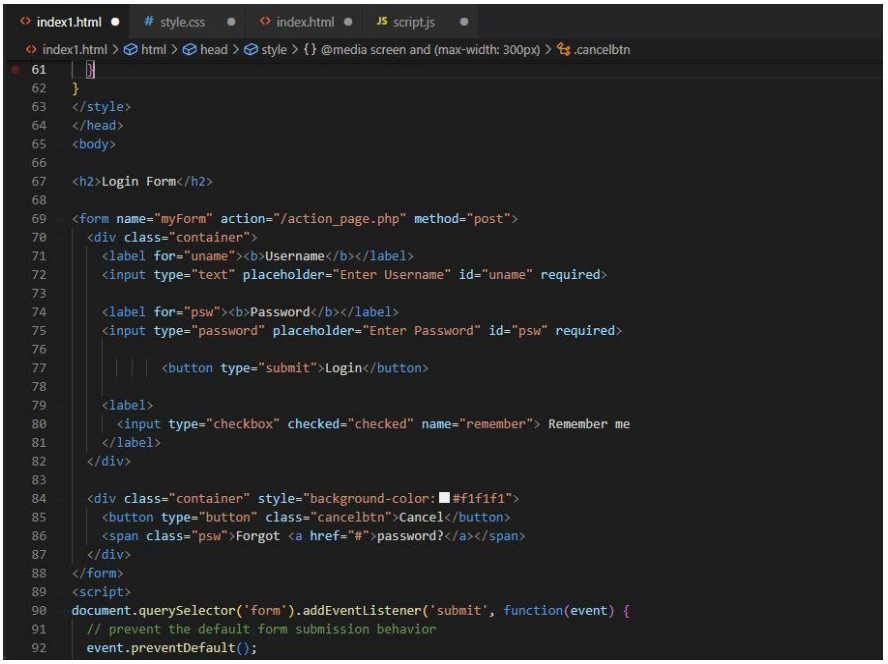
```
  padding: 8px 15px;
}
form button{
  width: 100%;
  height: 50px;
  color: #fff;
  outline: none;
  border: none;
  cursor: pointer;
  font-size: 17px;
  border-radius: 4px;
  background: #6A93F8;
}
@media (max-width: 660px){
  .wrapper{
    margin: 15px;
    gap: 15px;
    grid-template-columns: repeat(auto-fill, 100%);
  }
  .popup-box .popup{
    max-width: calc(100% - 15px);
  }
  .bottom-content .settings i{
    font-size: 17px;
  }
}
```

index1.html

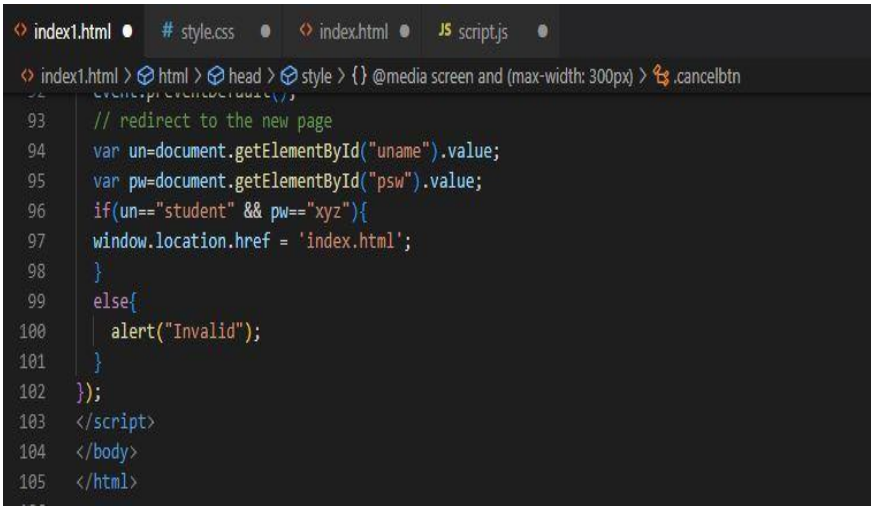




```
index1.html ● # style.css ● index.html ● JS script.js ●
index1.html > html > head > style > button
    background-color: #f44336;
  }

  .imgcontainer {
    text-align: center;
    margin: 24px 0 12px 0;
  }


  .container {
    padding: 16px;
  }

  span.psw {
    float: right;
    padding-top: 16px;
  }

  /* Change styles for span and cancel button on extra small screens */
  @media screen and (max-width: 300px) {
    span.psw {
       display: block;
       float: none;
    }
    .cancelbtn {
       width: 100%;
    }
  }
  </style>
  </head>
  <body>

  <h2>Login Form</h2>
```



```
index1.html ● # style.css ● index.html ● JS script.js ●
index1.html > html > head > style > {} @media screen and (max-width: 300px) > .cancelbtn
    }
  }
  </style>
  </head>
  <body>

  <h2>Login Form</h2>

  <form name="myForm" action="/action_page.php" method="post">
    <div class="container">
      <label for="uname"><b>Username</b></label>
      <input type="text" placeholder="Enter Username" id="uname" required>

      <label for="psw"><b>Password</b></label>
      <input type="password" placeholder="Enter Password" id="psw" required>

          <button type="submit">Login</button>

      <label>
        <input type="checkbox" checked="checked" name="remember"> Remember me
      </label>
    </div>

    <div class="container" style="background-color: #f1f1f1">
      <button type="button" class="cancelbtn">Cancel</button>
      <span class="psw">Forgot <a href="#">password?</a></span>
    </div>
  </form>
  <script>
  document.querySelector('form').addEventListener('submit', function(event) {
    // prevent the default form submission behavior
    event.preventDefault();
```



```
index1.html ● # style.css ● index.html ● JS script.js ●
index1.html > html > head > style > {} @media screen and (max-width: 300px) > .cancelbtn
    // redirect to the new page
    var un=document.getElementById("uname").value;
    var pw=document.getElementById("psw").value;
    if(un=="student" && pw=="xyz"){
    window.location.href = 'index.html';
    }
    else{
      alert("Invalid");
    }
  });
  </script>
  </body>
  </html>
```

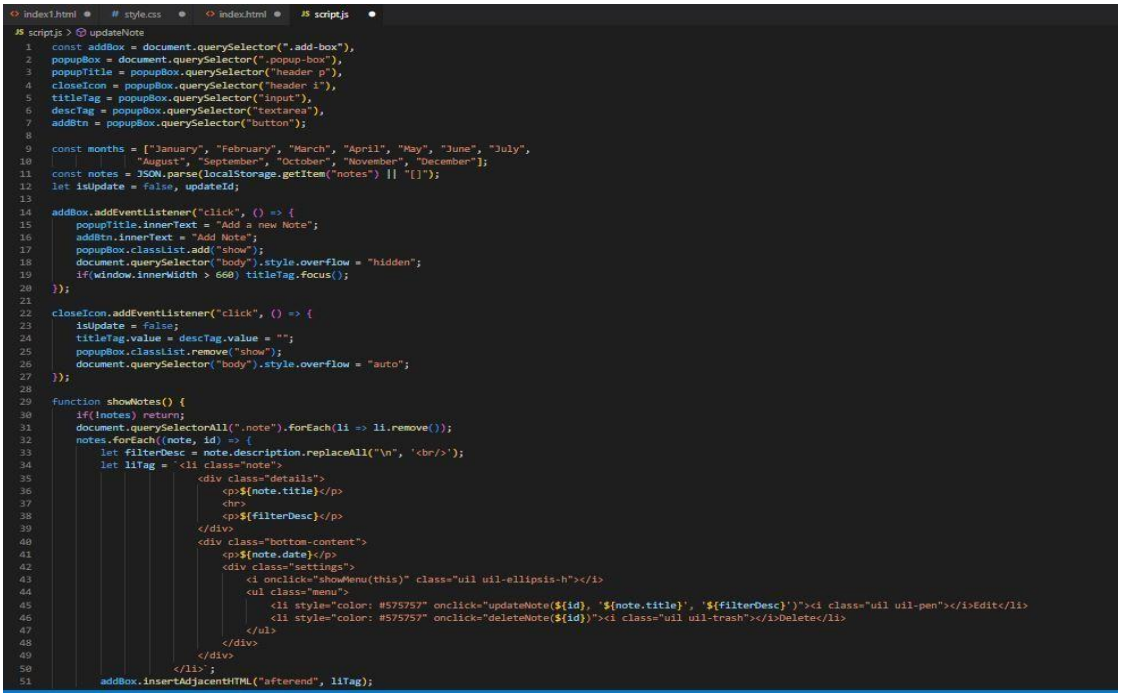
index.html

index1.html | style.css | index.html | script.js

index.html > html > body > div.popup-box > div.popup > div.content > form > div.row.title

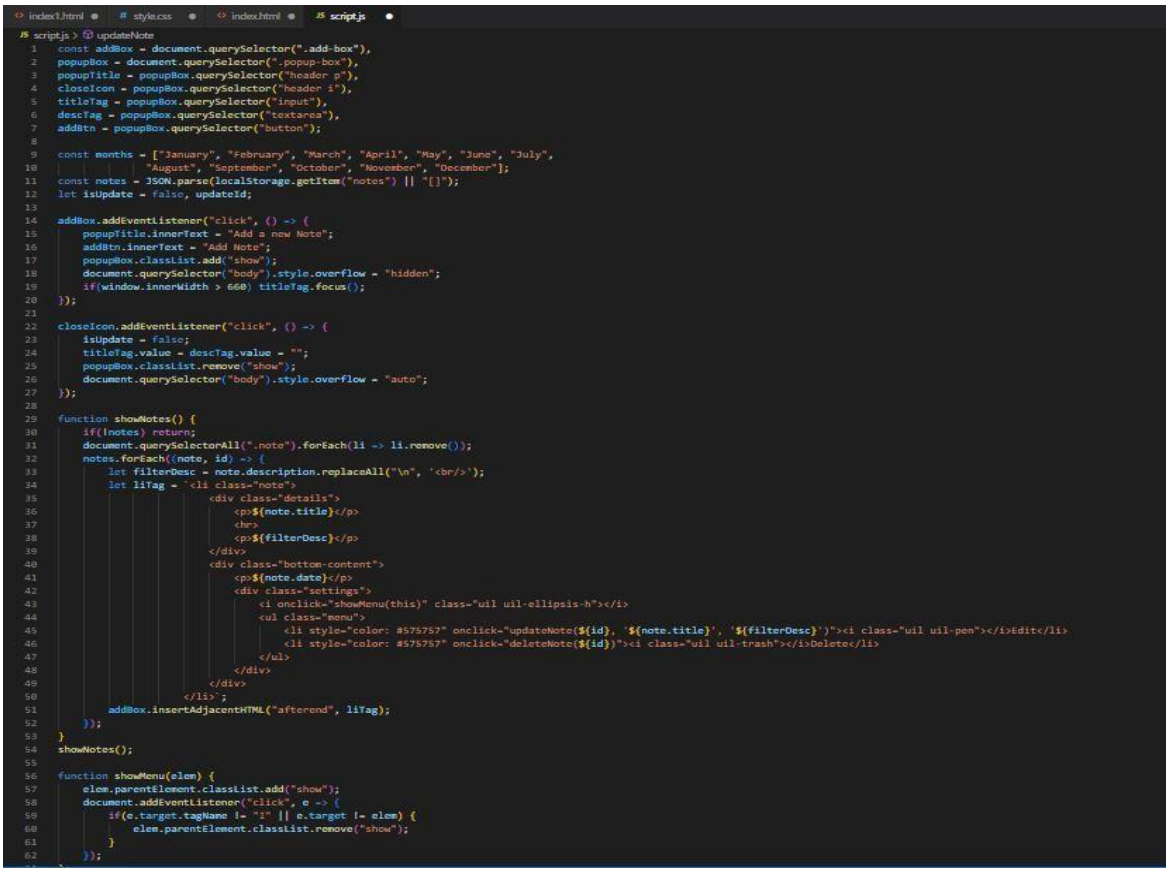
```
<!DOCTYPE html>
<!-- Coding By CodingNepal - youtube.com/codingnepal -->
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <title>Notes App in JavaScript | CodingNepal</title>
    <link rel="stylesheet" href="style.css">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <!-- Iconscout Link For Icons -->
    <link rel="stylesheet" href="https://unicons.iconscout.com/release/v4.0.0/css/line.css">
  </head>
  <body>
    <div style="text-align: center; background-color: #575757; color: #fff;">
      <h1 style="padding-left: -40%;">My Note App</h1>
      <p>make a note</p>
    </div>
    <div class="popup-box">
      <div class="popup">
        <div class="content">
          <header>
            <p></p>
            <i class="uil uil-times"></i>
          </header>
          <form action="#">
            <div class="row title">
              <label>Title</label>
              <input type="text" spellcheck="false">
            </div>
            <div class="row description">
              <label>Description</label>
              <textarea spellcheck="false"></textarea>
            </div>
            <button></button>
          </form>
        </div>
      </div>
    </div>
    <div class="wrapper">
      <li class="add-box">
        <div class="icon"><i class="uil uil-plus"></i></div>
        <p>Add new note</p>
      </li>
    </div>
    <script src="script.js"></script>

  </body>
</html>
```

script.js

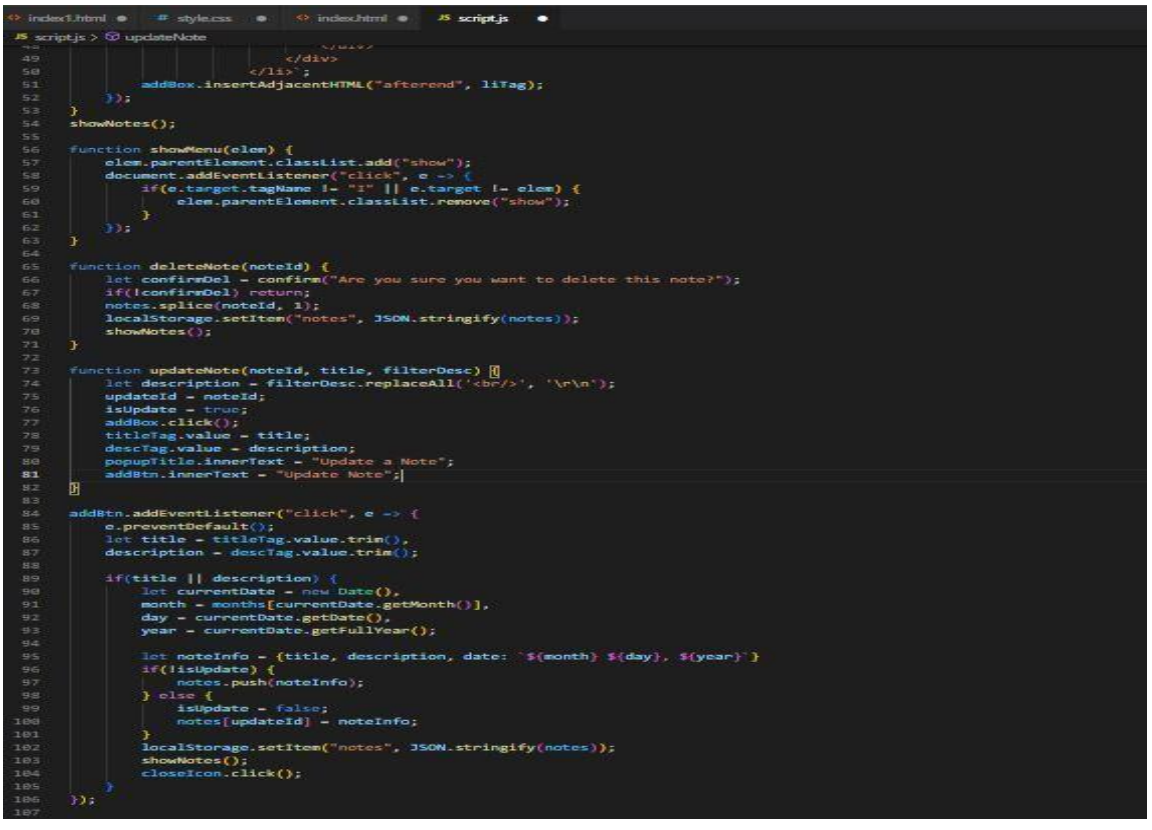


```
const addBox = document.querySelector(".add-box"),
popupBox = document.querySelector(".popup-box"),
popupTitle = popupBox.querySelector("header p"),
closeIcon = popupBox.querySelector("header i"),
titleTag = popupBox.querySelector("input"),
descTag = popupBox.querySelector("textarea"),
addBtn = popupBox.querySelector("button");

const months = ["January", "February", "March", "April", "May", "June", "July",
                "August", "September", "October", "November", "December"];
const notes = JSON.parse(localStorage.getItem("notes") || "[]");
let isUpdate = false, updateId;

addBox.addEventListener("click", () => {
    popupTitle.innerText = "Add a new Note";
    addBtn.innerText = "Add Note";
    popupBox.classList.add("show");
    document.querySelector("body").style.overflow = "hidden";
    if(window.innerWidth > 660) titleTag.focus();
});

closeIcon.addEventListener("click", () => {
    isUpdate = false;
    titleTag.value = descTag.value = "";
    popupBox.classList.remove("show");
    document.querySelector("body").style.overflow = "auto";
});

function showNotes() {
    if(!notes) return;
    document.querySelectorAll(".note").forEach(li => li.remove());
    notes.forEach((note, id) => {
        let filterDesc = note.description.replaceAll("\n", '<br/>');
        let liTag = `<li class="note">
                        <div class="details">
                            <p>${note.title}</p>
                            <hr>
                            <p>${filterDesc}</p>
                        </div>
                        <div class="bottom-content">
                            <p>${note.date}</p>
                            <div class="settings">
                                <i onclick="showMenu(this)" class="uil uil-ellipsis-h"></i>
                                <ul class="menu">
                                    <li style="color: #575757" onclick="updateNote(${id}, '${note.title}', '${filterDesc}')"><i class="uil uil-pen"></i>Edit</li>
                                    <li style="color: #575757" onclick="deleteNote(${id})"><i class="uil uil-trash"></i>Delete</li>
                                </ul>
                            </div>
                        </div>
                    </li>`;
        addBox.insertAdjacentHTML("afterend", liTag);
    });
}
showNotes();

function showMenu(elem) {
    elem.parentElement.classList.add("show");
    document.addEventListener("click", e => {
        if(e.target.tagName != "I" || e.target != elem) {
            elem.parentElement.classList.remove("show");
        }
    });
}

function deleteNote(noteId) {
    let confirmDel = confirm("Are you sure you want to delete this note?");
    if(!confirmDel) return;
    notes.splice(noteId, 1);
    localStorage.setItem("notes", JSON.stringify(notes));
    showNotes();
}

function updateNote(noteId, title, filterDesc) {
    let description = filterDesc.replaceAll('<br/>', '\r\n');
    updateId = noteId;
    isUpdate = true;
    addBox.click();
    titleTag.value = title;
    descTag.value = description;
    popupTitle.innerText = "Update a Note";
    addBtn.innerText = "Update Note";
}

addBtn.addEventListener("click", e => {
    e.preventDefault();
    let title = titleTag.value.trim(),
    description = descTag.value.trim();

    if(title || description) {
        let currentDate = new Date(),
        month = months[currentDate.getMonth()],
        day = currentDate.getDate(),
        year = currentDate.getFullYear();

        let noteInfo = {title, description, date: `${month} ${day}, ${year}`}
        if(!isUpdate) {
            notes.push(noteInfo);
        } else {
            isUpdate = false;
            notes[updateId] = noteInfo;
        }
        localStorage.setItem("notes", JSON.stringify(notes));
        showNotes();
        closeIcon.click();
    }
});
```

style.css



```
index1.html ● # style.css ● index.html ● JS script.js ●
JS script.js > updateNote
const addBox = document.querySelector(".add-box"),
popupBox = document.querySelector(".popup-box"),
popupTitle = popupBox.querySelector("header p"),
closeIcon = popupBox.querySelector("header i"),
titleTag = popupBox.querySelector("input"),
descTag = popupBox.querySelector("textarea"),
addBtn = popupBox.querySelector("button");

const months = ["January", "February", "March", "April", "May", "June", "July",
                "August", "September", "October", "November", "December"];
const notes = JSON.parse(localStorage.getItem("notes") || "[]");
let isUpdate = false, updateId;

addBox.addEventListener("click", () => {
    popupTitle.innerText = "Add a new Note";
    addBtn.innerText = "Add Note";
    popupBox.classList.add("show");
    document.querySelector("body").style.overflow = "hidden";
    if(window.innerWidth > 660) titleTag.focus();
});

closeIcon.addEventListener("click", () => {
    isUpdate = false;
    titleTag.value = descTag.value = "";
    popupBox.classList.remove("show");
    document.querySelector("body").style.overflow = "auto";
});

function showNotes() {
    if(!notes) return;
    document.querySelectorAll(".note").forEach(li => li.remove());
    notes.forEach((note, id) => {
        let filterDesc = note.description.replaceAll("\n", '<br/>');
        let liTag = `<li class="note">
                        <div class="details">
                            <p>${note.title}</p>
                            <br>
                            <p>${filterDesc}</p>
                        </div>
                        <div class="bottom-content">
                            <p>${note.date}</p>
                            <div class="settings">
                                <i onclick="showMenu(this)" class="uil uil-ellipsis-h"></i>
                                <ul class="menu">
                                    <li style="color: #575757" onclick="updateNote(${id}, '${note.title}', '${filterDesc}')"><i class="uil uil-pen"></i>Edit</li>
                                    <li style="color: #575757" onclick="deleteNote(${id})"><i class="uil uil-trash"></i>Delete</li>
                                </ul>
                            </div>
                        </div>
                    </li>`;
        addBox.insertAdjacentHTML("afterend", liTag);
    });
}
showNotes();

function showMenu(elem) {
    elem.parentElement.classList.add("show");
    document.addEventListener("click", e => {
        if(e.target.tagName != "I" || e.target != elem) {
            elem.parentElement.classList.remove("show");
        }
    });
}
```



```
index1.html ● # style.css ● index.html ● JS script.js ●
JS script.js > updateNote
                        </div>
                    </li>`;
        addBox.insertAdjacentHTML("afterend", liTag);
    });
}
showNotes();

function showMenu(elem) {
    elem.parentElement.classList.add("show");
    document.addEventListener("click", e => {
        if(e.target.tagName != "I" || e.target != elem) {
            elem.parentElement.classList.remove("show");
        }
    });
}

function deleteNote(noteId) {
    let confirmDel = confirm("Are you sure you want to delete this note?");
    if(!confirmDel) return;
    notes.splice(noteId, 1);
    localStorage.setItem("notes", JSON.stringify(notes));
    showNotes();
}

function updateNote(noteId, title, filterDesc) {
    let description = filterDesc.replaceAll('<br/>', '\r\n');
    updateId = noteId;
    isUpdate = true;
    addBox.click();
    titleTag.value = title;
    descTag.value = description;
    popupTitle.innerText = "Update a Note";
    addBtn.innerText = "Update Note";
}

addBtn.addEventListener("click", e => {
    e.preventDefault();
    let title = titleTag.value.trim(),
    description = descTag.value.trim();

    if(title || description) {
        let currentDate = new Date(),
        month = months[currentDate.getMonth()],
        day = currentDate.getDate(),
        year = currentDate.getFullYear();

        let noteInfo = {title, description, date: `${month} ${day}, ${year}`}
        if(!isUpdate) {
            notes.push(noteInfo);
        } else {
            isUpdate = false;
            notes[updateId] = noteInfo;
        }
        localStorage.setItem("notes", JSON.stringify(notes));
        showNotes();
        closeIcon.click();
    }
});
```

2. Output

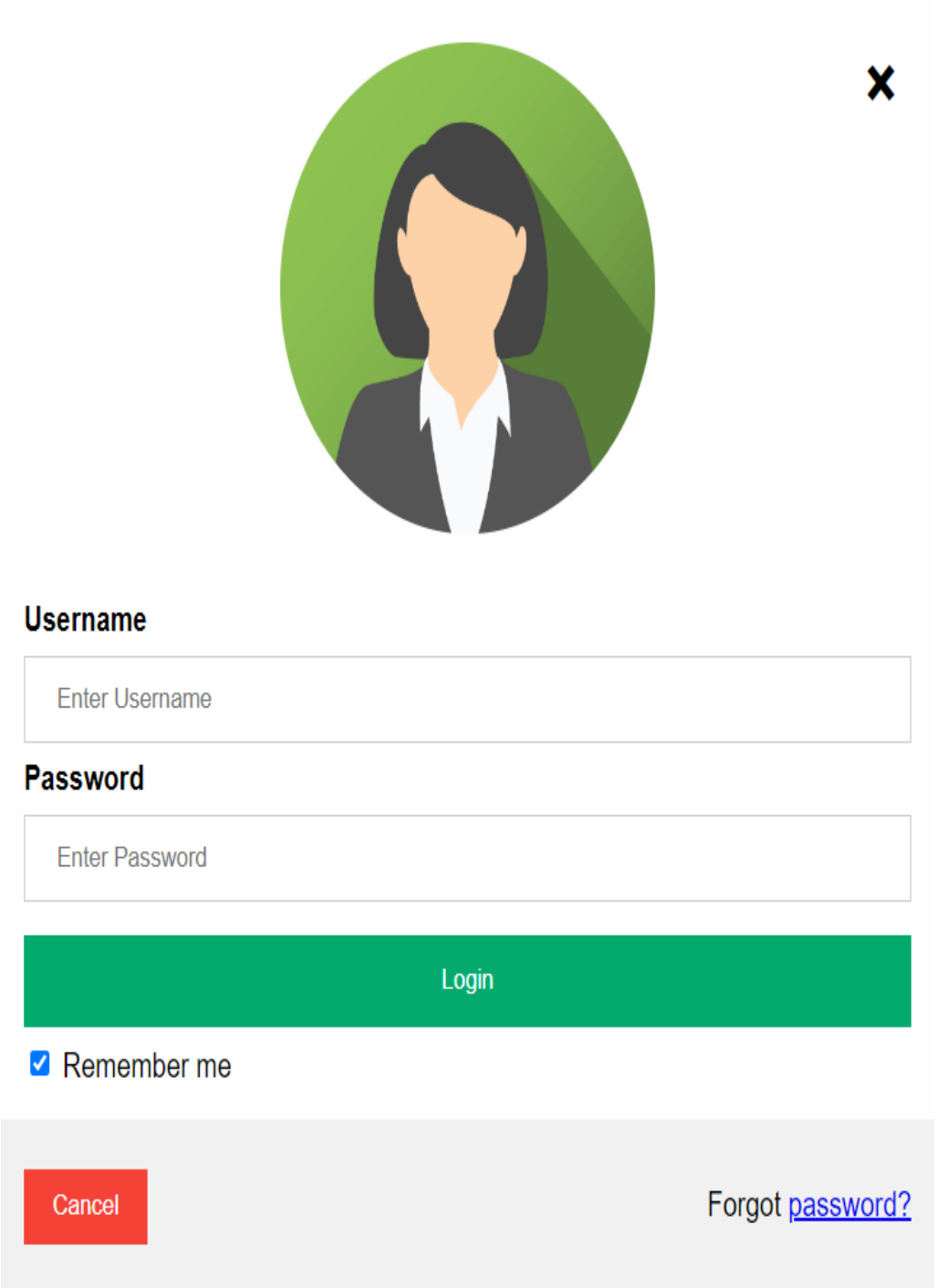


Fig 1: Login Page

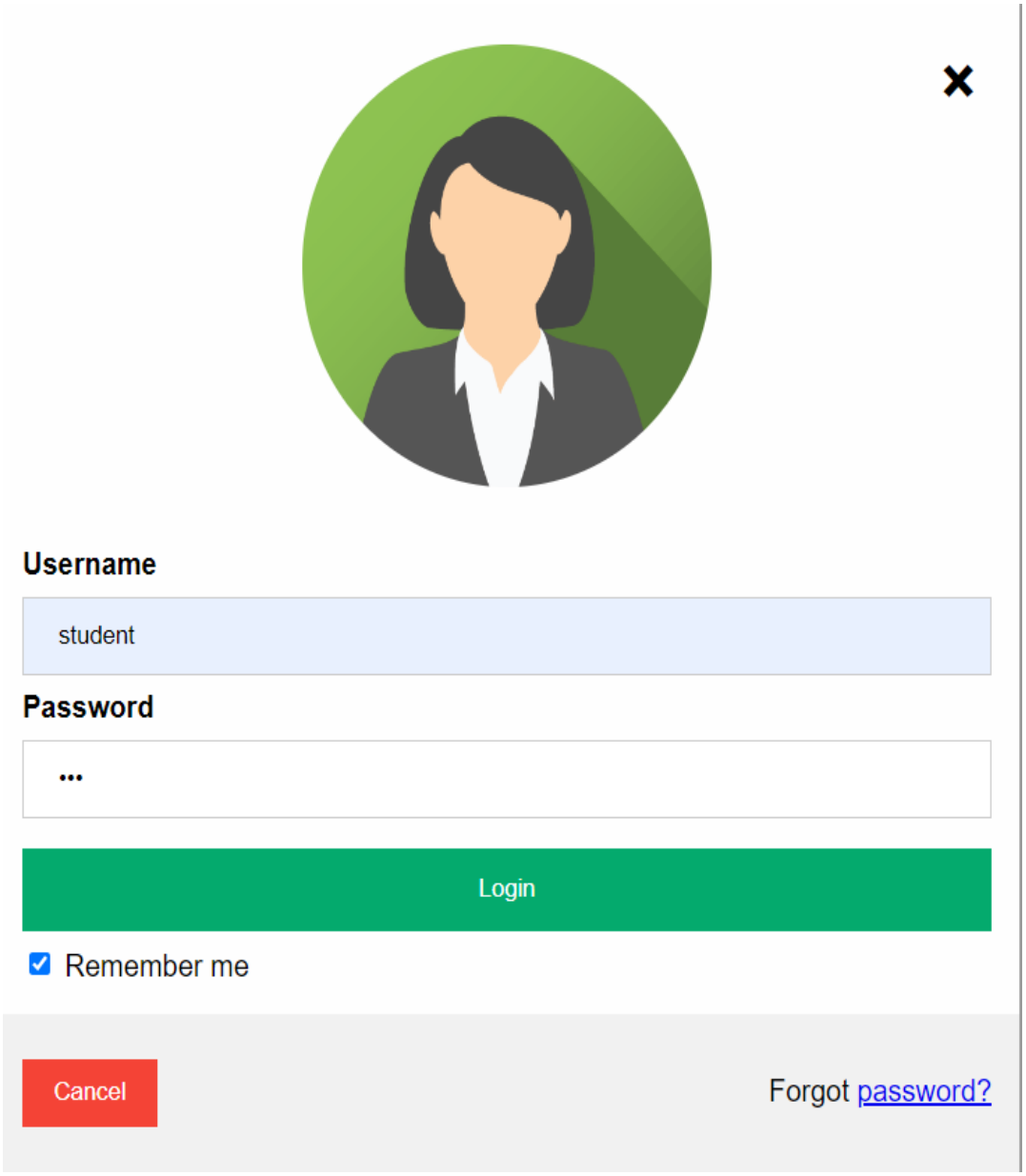


Fig 2: Login page wirh user credentials

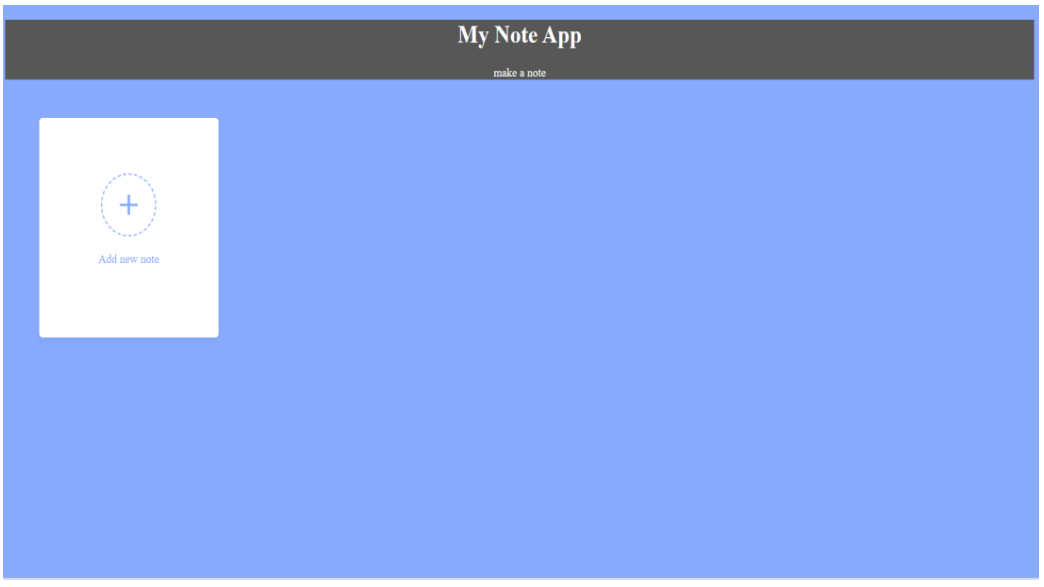


Fig 3: To create a note

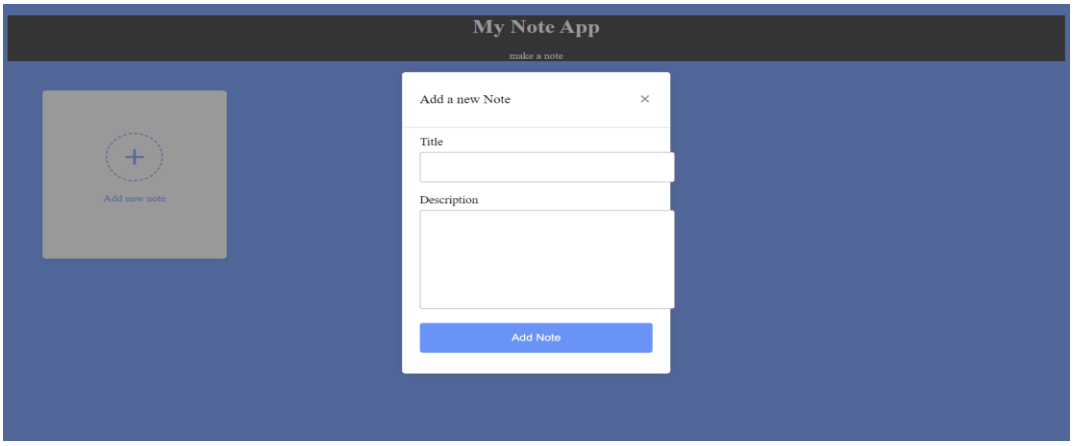


Fig 4: To add title and description of the note

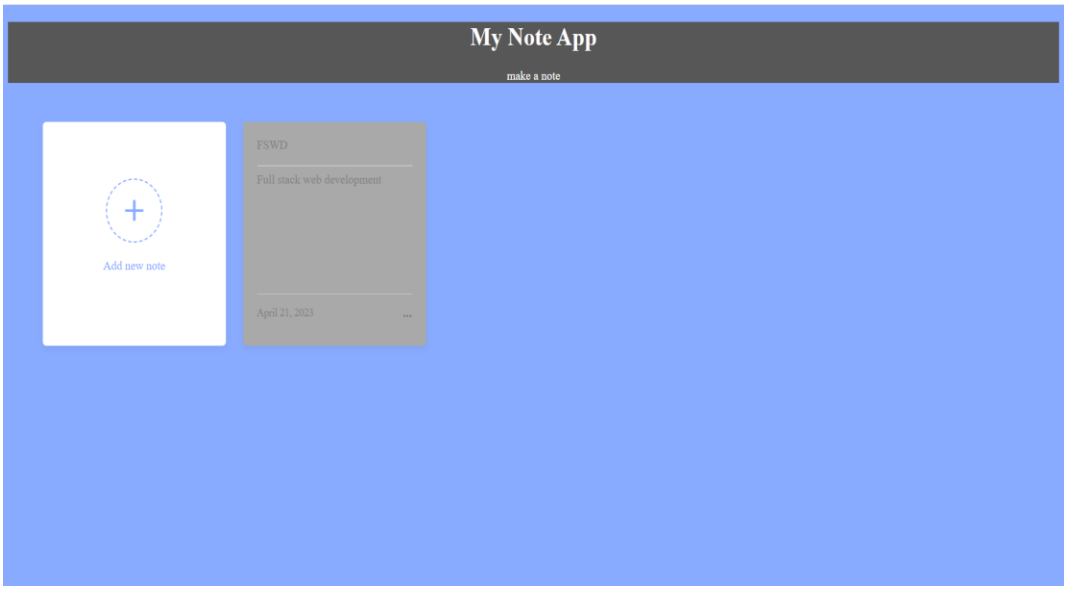


Fig 5: Note created

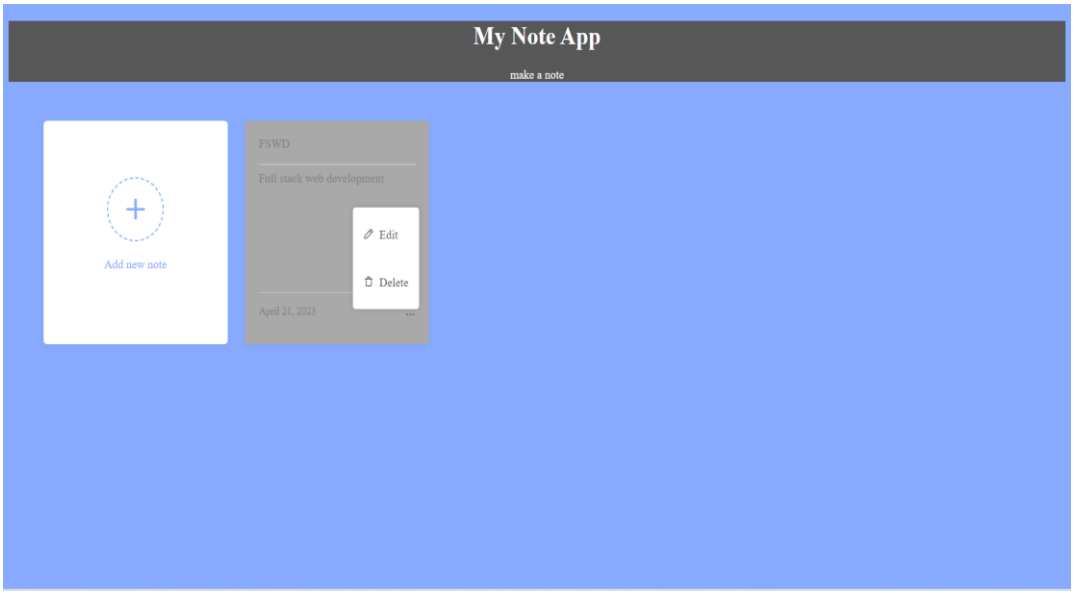


Fig 6: To edit and delete the note

3. Conclusion

Overall, MyNote provides a simple and efficient way to manage schedules, notes, to-do lists, shopping lists, and other important information. The app is user-friendly, easy to use, and helps to streamline the note-taking process, making it an excellent tool for anyone who needs to take notes regularly. Making notes in mobile applications provides several benefits, including convenience, organization, accessibility, collaboration, and security. Mobile applications are an excellent tool for anyone who needs to take notes regularly, making the process easier and more efficient.

4. References

- https://stackoverflow.com/
- https://www.w3schools.com/
- https://www.youtube.com/
- https://www.freecodecamp.org/
- https://www.simplilearn.com/tutorials/