

## EXP:1

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>To Do-List</title>
  <style>
    /* General CSS */
    html, body {
      height: 100%;
    }
    html, body, .container {
      margin: 0; padding: 0;
      width: 100%;
      font-family: Roboto, Arial;
    }
    body {
      background-image: url(bckgrnd.png); background-size: cover;
      display: flex; align-items: center; justify-content: center;
    }

    /* Container CSS */
    .container {
      margin-top: 20px; padding-bottom: 12px;
      width: 80%; max-width: 800px;
      background-color: rgba(126, 126, 126, 0.904);
      display: flex; flex-direction: column;
      border-radius: 40px;
    }

    /* Main CSS */

    /* Upper-Div */
    .upper-div {
      display: grid; place-items: center; grid-template-rows: 6fr 10fr;
    }

    .name-area {
      display: flex;
      align-items: center;
      justify-content: center;
    }
    .header {
      font-size: 42px; font-weight: 600;
```

```

    margin: 12px 0;
}

#name {
    padding: 10px 40px 10px 20px;
    border-radius: 12px; border: 2.5px solid grey;
    font-size: 15px; font-weight: 600;
}

.list-adder {
    padding: 10px 40px 10px 20px;
    border-radius: 12px; border: 2.5px solid grey;
    font-size: 15px; font-weight: 600;
}

.adder {
    padding: 10px 15px;
    color: aliceblue; background-color: black;
    border-radius: 6px; border: none;
    font-size: 16px; font-weight: 600;
    transition: all 0.8s;
}

.adder:hover {
    cursor: pointer;
}

.adder:active, .adder:focus {
    color: black; background-color: white;
    border: none;
}

/* Lower-Div */
.lower-con {
    height: 100%;
    display: flex; flex-direction: column; row-gap: 18px; align-items: center; justify-content:
center;
}

.content-area {
    display: flex; flex-direction: row; column-gap: 18px;
    margin: 0px 25px;
    text-align: center;
}

.check-box {
    padding: 11px 0px;
    font-size: 24px;
    cursor: pointer;
}

```

```

.text-area {
  font-size: 18px; text-decoration: none;
}
.cross-btn {
  position: relative; top: 10px; cursor: pointer;
  width: 36px; height: 36px;
  font-size: 20px; color: black; background-color: transparent;
  border: none; border-radius: 50px;
}
.cross-btn:hover {
  border-radius: 50px;
  background-color: rgb(213, 213, 213);
}
h2 {
  font-size: 40px;
}
</style>
</head>
<body>
<div class="container">
  <div class="upper-div">
    <div class="header">
      To-Do List
    </div>
    <div class="Adder-con">
      <select name="name" id="name">
        <option value="joe">JOE</option>
        <option value="ram">RAM</option>
        <option value="ravi">RAVI</option>
        <option value="renu">RENU</option>
      </select>
      <input type="text" class="list-adder">
      <button class="add" >Add</button>
    </div>
  </div>
  <div class="lower-con">
    <h2>Your Plan List</h2>
  </div>
</div>
<script>
// Getting elements
const inpbx = document.querySelector('.list-adder');
const addbtn = document.querySelector('.add');
const nameSelect = document.querySelector('#name');

```

```

const lowerDiv = document.querySelector('.lower-con');
//Variable
var idno = 0;

// Event Listeners
addbtn.addEventListener('click', adder);
inpbx.addEventListener('keypress', function (e) {
  if (e.key === 'Enter') {
    adder();
  }
});

//Functions
function adder() {
  idno++;
  if (inpbx.value === "") {
    alert("Enter Your Plan");
  } else {
    var value = inpbx.value;
    var name = nameSelect.value.toUpperCase();
    lowerDiv.innerHTML += `
      <div class="content-area" id="ca-${idno}">

        <span class="check-box" onclick="Striker(${idno})" id="cb-${idno}"
ondblclick="nonStriker(${idno})">&#9744;</span>
        <span class="name-area">${name} - </span>
        <p class="text-area" onclick="Striker(${idno})" id="ta-${idno}">${value}</p>
        <button class="cross-btn" onclick="remover(${idno})">&#x2715;</button>
      </div>
    `;
  }
  inpbx.value = "";
  setter();
  console.log(getter());
}

function Striker(idNo) {
  const text = document.getElementById(`ta-${idNo}`);
  const check = document.getElementById(`cb-${idNo}`);
  text.style = 'text-decoration:line-through';
  check.innerHTML = '&#9745;';
  setter();
}

```

```

function nonStriker(idNo) {
  const text = document.getElementById(`ta-${idNo}`);
  const check = document.getElementById(`cb-${idNo}`);
  text.style = 'text-decoration:none';
  check.innerHTML = '&#9744;';
  setter();
}

function remover(idNo) {
  const content = document.getElementById(`ca-${idNo}`);
  content.remove();
  setter();
}

function setter() {
  localStorage.setItem("Data", lowerDiv.innerHTML);
}

function getter() {
  lowerDiv.innerHTML = localStorage.getItem("Data");
}

getter();
</script>
</body>
</html>

```

## EXP:2

### HTML

```

<!DOCTYPE html>
<html>
<head>
  <title>Login Form</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      background-color: #f0f0f0;
      display: flex;
      justify-content: center;
      align-items: center;
      height: 100vh;
    }
  </style>

```

```
    margin: 0;
}

h2 {
    margin-bottom: 20px;
}

form {
    background-color: #fff;
    padding: 20px;
    border-radius: 5px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
    width: 300px;
}

div {
    margin-bottom: 15px;
}

label {
    display: block;
    margin-bottom: 5px;
}

input[type="text"],
input[type="password"] {
    width: 100%;
    padding: 8px;
    border: 1px solid #ccc;
    border-radius: 3px;
}

button {
    width: 100%;
    padding: 10px;
    background-color: #007bff;
    border: none;
    color: white;
    border-radius: 3px;
    cursor: pointer;
}

button:hover {
    background-color: #0056b3;
}
</style>
```

```

    <script src="script.js" defer></script>
</head>
<body>
  <h2>Login Form</h2>
  <form id="loginForm">
    <div>
      <label for="username">Username:</label>
      <input type="text" id="username" name="username">
    </div>
    <br>
    <div>
      <label for="password">Password:</label>
      <input type="password" id="password" name="password">
    </div>
    <br>
    <button type="submit">Login</button>
  </form>
</body>
</html>

```

## JS

```

document.addEventListener('DOMContentLoaded', function() {
  document.getElementById('loginForm').onsubmit = function() {
    var username = document.getElementById("username").value;
    var password = document.getElementById("password").value;

    var staticUsername = "user";
    var staticPassword = "password123";

    if (username === staticUsername && password === staticPassword) {
      window.location.href = "nextpage.html";
    } else {
      alert("Invalid username or password!");
    }

    return false;
  }
});

```

## EXP:3

```

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

```

```
const port = 3000;
// Parse JSON bodies for POST and PUT requests
app.use(bodyParser.json());
let items = [
  { id: 1, name: 'Item 1', quantity: 1 },
  { id: 2, name: 'Item 2', quantity: 2 },
  { id: 3, name: 'Item 3', quantity: 3 },
];
// GET all items
app.get('/api/items', (req, res) => {
  res.json(items);
});
// GET single item by ID
app.get('/api/items/:id', (req, res) => {
  const id = parseInt(req.params.id);
  const item = items.find(item => item.id === id);
  if (!item) return res.status(404).send('Item not found');
  res.json(item);
});
// POST new item
app.post('/api/items', (req, res) => {
  const newItem = req.body;
  items.push(newItem);
  res.status(201).json(newItem);
});
// PUT update item by ID
app.put('/api/items/:id', (req, res) => {
  const id = parseInt(req.params.id);
  const updatedItem = req.body;
  const index = items.findIndex(item => item.id === id);
  if (index === -1) return res.status(404).send('Item not found');
  items[index] = updatedItem;
  res.json(updatedItem);
});
// DELETE item by ID
app.delete('/api/items/:id', (req, res) => {
  const id = parseInt(req.params.id);
  items = items.filter(item => item.id !== id);
  res.sendStatus(204);
});
// PATCH update item partially by ID
app.patch('/api/items/:id', (req, res) => {
  const id = parseInt(req.params.id);
  const updatedFields = req.body;
```



```

    const index = items.findIndex(item => item.id === id);
    if (index === -1) return res.status(404).send('Item not found');
    items[index] = { ...items[index], ...updatedFields };
    res.json(items[index]);
  });
  // Start the server
  app.listen(port, () => {
    console.log(`Server running at http://localhost:${port}`);
  });

```

#### EXP:4

```

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Chat Between User 1 and User 2</title>
<link rel="stylesheet" href="styles.css">
<style>
body {
font-family: Arial, sans-serif;
}
.chat-container {
display: flex;
justify-content: space-between;
margin: 20px auto;
max-width: 800px;
}
.user {
flex: 1;
border: 10px solid black;
border-radius: 10px;
padding: 10px;
margin-right: 20px; /* Add margin-right for space between user divs */
}
.chat-box {
height: 200px;
overflow-y: auto;
margin-bottom: 10px;
padding: 10px;
border: 1px solid black; /* Add border for the message box */
border-radius: 5px; /* Add border radius for the message box */
}

```

```

}
.message-input {
width: calc(100%- 70px);
padding: 5px;
margin-right: 5px;
border: 1px solid #ccc;
border-radius: 5px;
outline: none;
}
.send-btn {
width: 60px;
padding: 5px;
cursor: pointer;
background-color: #4CAF50;
color: white;
border: none;
border-radius: 5px;
text-align: center;
text-decoration: none;
display: inline-block;
font-size: 16px;
}
.message.sent {
text-align: left;
}
.message.received {
text-align: right;
}
.message.sent::before {
content: "User 1: ";
}
.message.received::before {
content: "User 2: ";
}
</style>
</head>
<body>
<div class="chat-container">
<div class="user" id="user1">
<h2>User 1</h2>
<div class="chat-box" id="user1-chat-box"></div>
<input type="text" class="message-input" id="user1-message-input" placeholder="Type
a message..."><br/><br/>
<button class="send-btn" id="user1-send-btn">Send</button>

```

```

</div>
</div>
<div class="user" id="user2">
<h2>User 2</h2>
<div class="chat-box" id="user2-chat-box"></div>
<input type="text" class="message-input" id="user2-message-input" placeholder="Type a
message..."><br/><br/>
<button class="send-btn" id="user2-send-btn">Send</button>
</div>
<script>
document.addEventListener("DOMContentLoaded", function() {
const user1ChatBox = document.getElementById("user1-chat-box");
const user1MessageInput = document.getElementById("user1-message-input");
const user1SendButton = document.getElementById("user1-send-btn");
const user2ChatBox = document.getElementById("user2-chat-box");
const user2MessageInput = document.getElementById("user2-message-input");
const user2SendButton = document.getElementById("user2-send-btn");
user1SendButton.addEventListener("click", function() {
sendMessage(user1MessageInput.value, user1ChatBox, user2ChatBox);
user1MessageInput.value = "";
});
user2SendButton.addEventListener("click", function() {
sendMessage(user2MessageInput.value, user2ChatBox, user1ChatBox);
user2MessageInput.value = "";
});
function sendMessage(message, senderChatBox, receiverChatBox) {
if (message.trim() !== "") {
displayMessage(message, senderChatBox, "sent");
// Display the message in the receiver's chat box
displayMessage(message, receiverChatBox, "received");
}
}
function displayMessage(message, chatBox, type) {
const messageElement = document.createElement("div");
messageElement.textContent = message;
messageElement.classList.add("message", type);
chatBox.appendChild(messageElement);
chatBox.scrollTop = chatBox.scrollHeight;
}
});
</script>
</body>
</html>

```