Web Development using MERN Stack

PRACTICAL FILE

GURU TEGH BAHADUR INSTITUTE OF TECHNOLOGY

Submitted By: Yashi Raj

Enrollment No.: 04976803121

Class-IT-3(FSD)

Submitted To: Ms. Kritika

Index

	complete/incomplete, and View weekly/monthly statistics of their to-dos.		
12	Blogging platform		
13	Social media platform		
14	Weather Forecasting App		
15	Bookstore Library and Stock-Keeping App		

Experiment – 3 a

Aim: To create a time-table using HTML

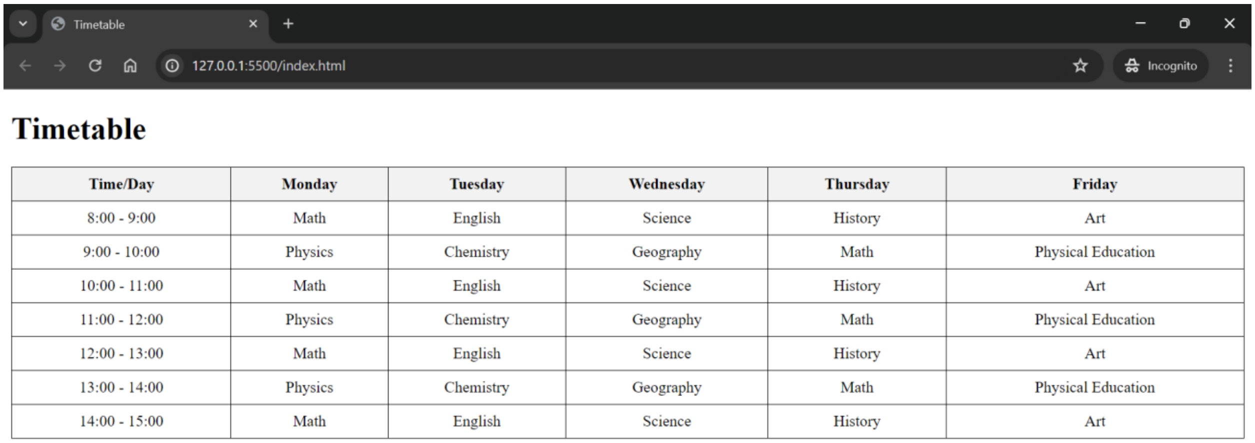
Code:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Timetable</title>
    <style>
        table {
            width: 100%;
            border-collapse: collapse;
        }
        th, td {
            border: 1px solid black;
            padding: 8px;
            text-align: center;
        }
        th {
            background-color: #f2f2f2;
        }
    </style>
</head>
<body>

<h1>Timetable</h1>

<table>
    <tr>
        <th>Time/Day</th>
        <th>Monday</th>
        <th>Tuesday</th>
        <th>Wednesday</th>
        <th>Thursday</th>
        <th>Friday</th>
    </tr>
    <tr>
        <td>8:00 - 9:00</td>
```

```
      <td>Math</td>
      <td>English</td>
      <td>Science</td>
      <td>History</td>
      <td>Art</td>
    </tr>
    <tr>
      <td>9:00 - 10:00</td>
      <td>Physics</td>
      <td>Chemistry</td>
      <td>Geography</td>
      <td>Math</td>
      <td>Physical Education</td>
    </tr>
    <tr>
      <td>10:00 - 11:00</td>
      <td>Math</td>
      <td>English</td>
      <td>Science</td>
      <td>History</td>
      <td>Art</td>
    </tr>
    <tr>
      <td>11:00 - 12:00</td>
      <td>Physics</td>
      <td>Chemistry</td>
      <td>Geography</td>
      <td>Math</td>
      <td>Physical Education</td>
    </tr>
    <tr>
      <td>12:00 - 13:00</td>
      <td>Math</td>
      <td>English</td>
      <td>Science</td>
      <td>History</td>
      <td>Art</td>
    </tr>
    <tr>
      <td>13:00 - 14:00</td>
      <td>Physics</td>
      <td>Chemistry</td>
      <td>Geography</td>
      <td>Math</td>
      <td>Physical Education</td>
```

```
  </tr>
  <tr>
    <td>14:00 - 15:00</td>
    <td>Math</td>
    <td>English</td>
    <td>Science</td>
    <td>History</td>
    <td>Art</td>
  </tr>
</table>
</body>
</html>
```

Output:



Timetable

Time/Day	Monday	Tuesday	Wednesday	Thursday	Friday
8:00 - 9:00	Math	English	Science	History	Art
9:00 - 10:00	Physics	Chemistry	Geography	Math	Physical Education
10:00 - 11:00	Math	English	Science	History	Art
11:00 - 12:00	Physics	Chemistry	Geography	Math	Physical Education
12:00 - 13:00	Math	English	Science	History	Art
13:00 - 14:00	Physics	Chemistry	Geography	Math	Physical Education
14:00 - 15:00	Math	English	Science	History	Art

Experiment – 3 b

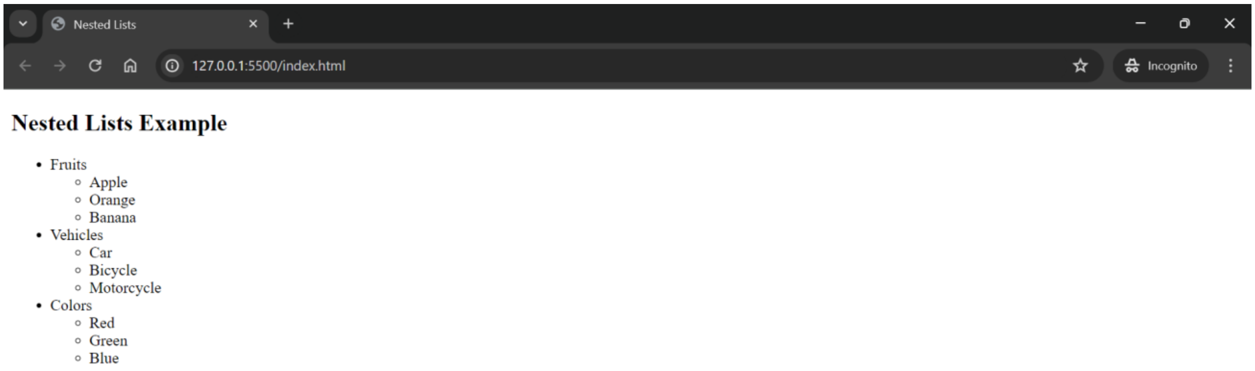
Aim: To create nested lists using HTML

Code:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Nested Lists</title>
</head>
<body>

<h2>Nested Lists Example</h2>

<ul>
    <li>Fruits
        <ul>
            <li>Apple</li>
            <li>Orange</li>
            <li>Banana</li>
        </ul>
    </li>
    <li>Vehicles
        <ul>
            <li>Car</li>
            <li>Bicycle</li>
            <li>Motorcycle</li>
        </ul>
    </li>
    <li>Colors
        <ul>
            <li>Red</li>
            <li>Green</li>
            <li>Blue</li>
        </ul>
    </li>
</ul>

</body>
```

```
</html>
```

Output:



Nested Lists Example

- Fruits
    - Apple
    - Orange
    - Banana
- Vehicles
    - Car
    - Bicycle
    - Motorcycle
- Colors
    - Red
    - Green
    - Blue

Experiment – 4 a

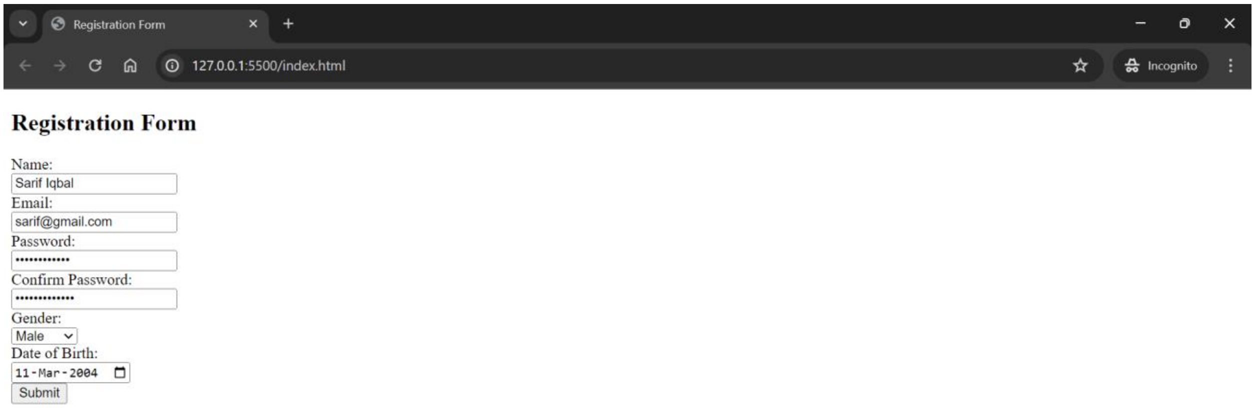
Aim: To create a registration form using HTML

Code:

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Registration Form</title>
</head>
<body>

<h2>Registration Form</h2>

<form action="/submit_registration" method="post">
    <label for="name">Name:</label><br>
    <input type="text" id="name" name="name" required><br>

    <label for="email">Email:</label><br>
    <input type="email" id="email" name="email" required><br>

    <label for="password">Password:</label><br>
    <input type="password" id="password" name="password" required><br>

    <label for="confirm_password">Confirm Password:</label><br>
    <input type="password" id="confirm_password" name="confirm_password" required><br>

    <label for="gender">Gender:</label><br>
    <select id="gender" name="gender">
        <option value="male">Male</option>
        <option value="female">Female</option>
        <option value="other">Other</option>
    </select><br>

    <label for="birthdate">Date of Birth:</label><br>
    <input type="date" id="birthdate" name="birthdate" required><br>

    <input type="submit" value="Submit">
</form>
```

```
</body>
</html>
```

Output:



Registration Form

Name:
Sarif Iqbal
Email:
sarif@gmail.com
Password:
••••••••••••
Confirm Password:
•••••••••••••
Gender:
Male
Date of Birth:
11-Mar-2004

Experiment – 4 b

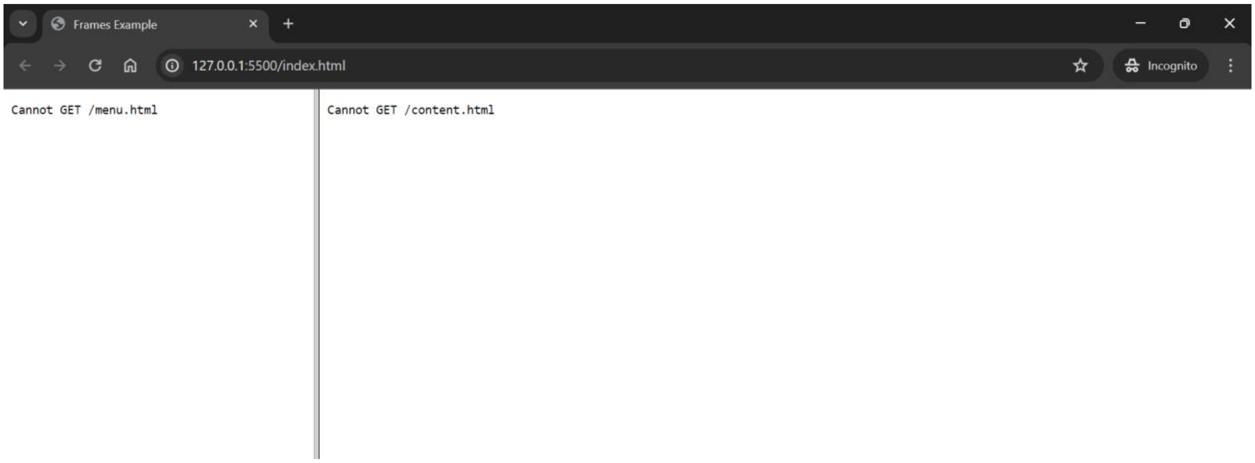
Aim: To create frames and hyperlinks using HTML

Code:

```
<!DOCTYPE html>
<html>
<head>
   <title>Frames Example</title>
</head>
<frameset cols="25%, 75%">
   <frame src="menu.html" name="menu">
   <frame src="content.html" name="content">
</frameset>
</html>
```

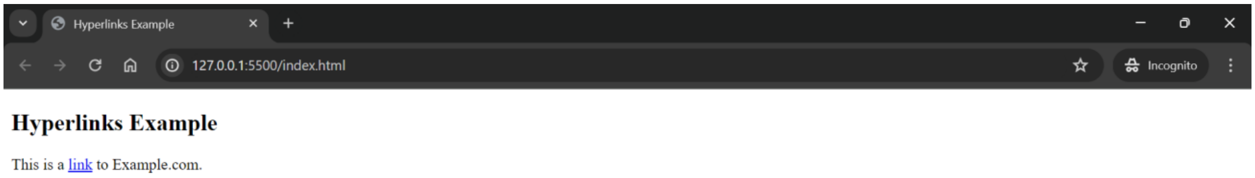
```
<!DOCTYPE html>
<html>
<head>
   <title>Hyperlinks Example</title>
</head>
<body>
   <h2>Hyperlinks Example</h2>
   <p>This is a <a href="https://www.example.com">link</a> to Example.com.</p>
</body>
</html>
```

Output:



```
Cannot GET /menu.html
```

```
Cannot GET /content.html
```



Hyperlinks Example

This is a link to Example.com.

Experiment – 5 a

Aim: Create a static website using HTML, CSS, and JavaScript

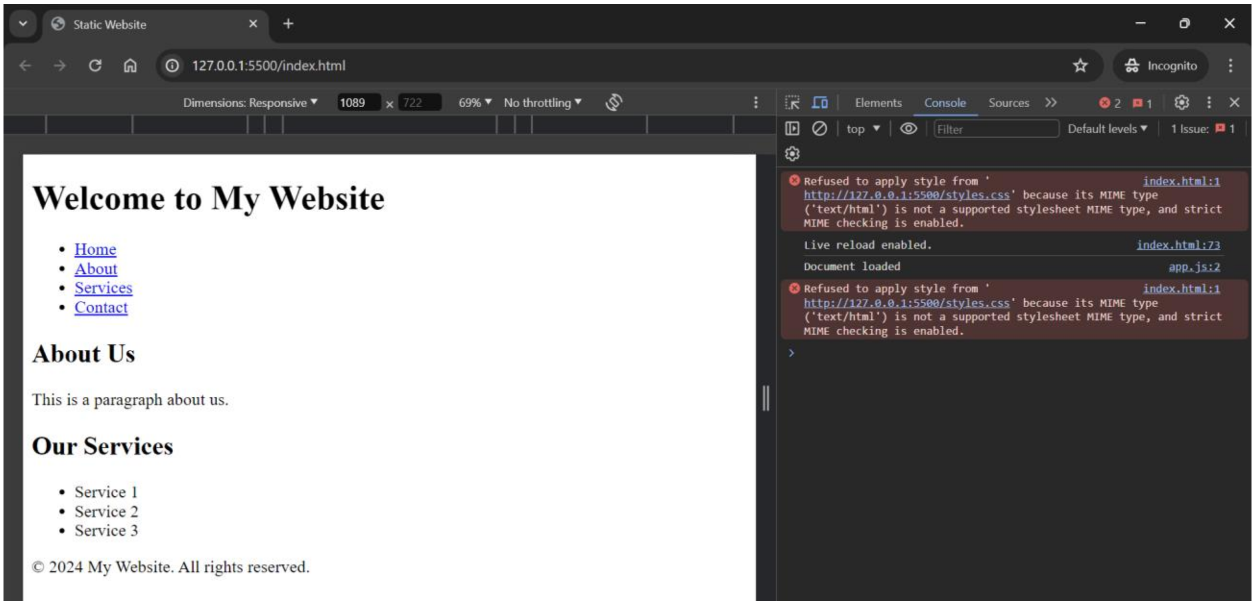
Code:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Static Website</title>
    <link rel="stylesheet" href="styles.css">
</head>
<body>

<header>
    <h1>Welcome to My Website</h1>
</header>

<nav>
    <ul>
        <li><a href="#">Home</a></li>
        <li><a href="#">About</a></li>
        <li><a href="#">Services</a></li>
        <li><a href="#">Contact</a></li>
    </ul>
</nav>

<main>
    <section>
        <h2>About Us</h2>
        <p>This is a paragraph about us.</p>
    </section>

    <section>
        <h2>Our Services</h2>
        <ul>
            <li>Service 1</li>
            <li>Service 2</li>
            <li>Service 3</li>
        </ul>
    </section>
</main>

<footer>
    <p>&copy; 2024 My Website. All rights reserved.</p>
</footer>

<script src="app.js"></script>
</body>
</html>
```

```
header {
    background-color: #333;
    color: #fff;
    padding: 20px;
    text-align: center;
}

nav ul {
    list-style-type: none;
    padding: 0;
    margin: 0;
}

nav ul li {
    display: inline;
    margin-right: 10px;
}

nav ul li a {
    text-decoration: none;
    color: #333;
}

main {
    padding: 20px;
}

section {
    margin-bottom: 20px;
}

footer {
    background-color: #333;
    color: #fff;
    padding: 10px 20px;
    text-align: center;
}
```

```
document.addEventListener("DOMContentLoaded", function() {
    console.log("Document loaded");
});
```

Output:



Welcome to My Website

- Home
- About
- Services
- Contact

About Us

This is a paragraph about us.

Our Services

- Service 1
- Service 2
- Service 3

© 2024 My Website. All rights reserved.

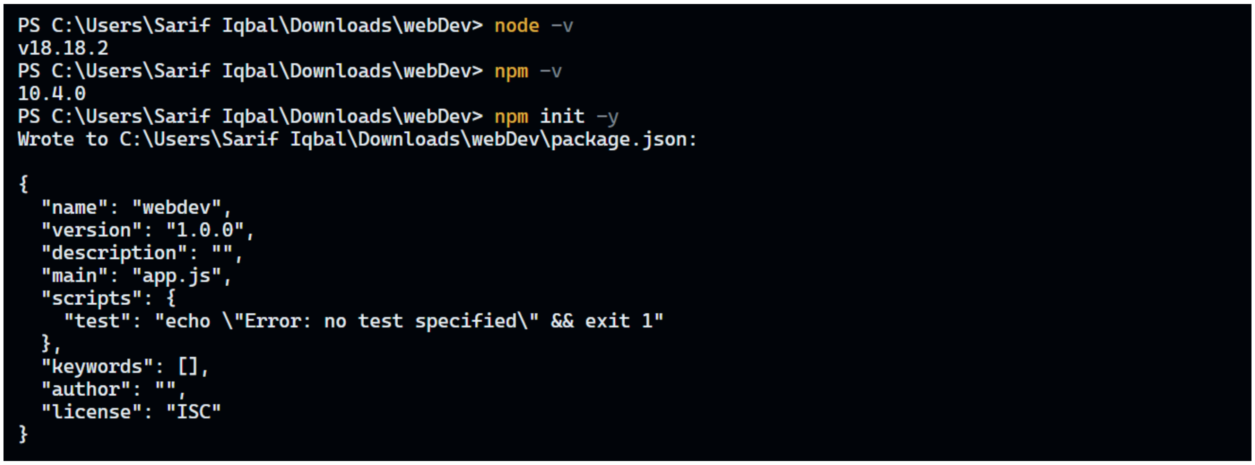
```
Refused to apply style from '                          index.html:1
http://127.0.0.1:5500/styles.css' because its MIME type
('text/html') is not a supported stylesheet MIME type, and strict
MIME checking is enabled.
Live reload enabled.                                  index.html:73
Document loaded                                           app.js:2
Refused to apply style from '                          index.html:1
http://127.0.0.1:5500/styles.css' because its MIME type
('text/html') is not a supported stylesheet MIME type, and strict
MIME checking is enabled.
>
```

Experiment – 5 b

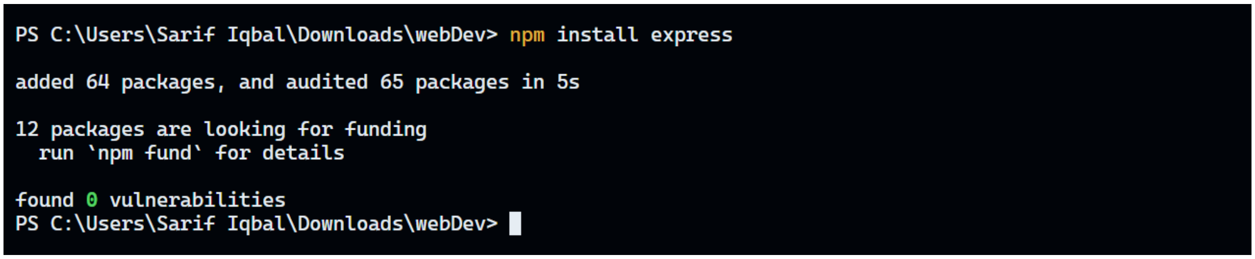
Aim: Install and set up Node.JS and Express.JS

Code:
```
const express = require('express');
const app = express();
const port = 3000;

app.get('/', (req, res) => {
  res.send('Hello World!');
});

app.listen(port, () => {
  console.log(`Server is running on http://localhost:${port}`);
});
```

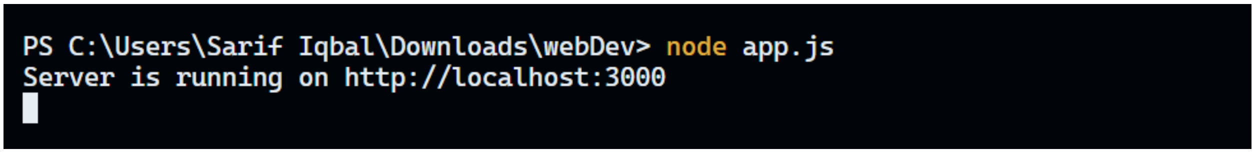
Output:



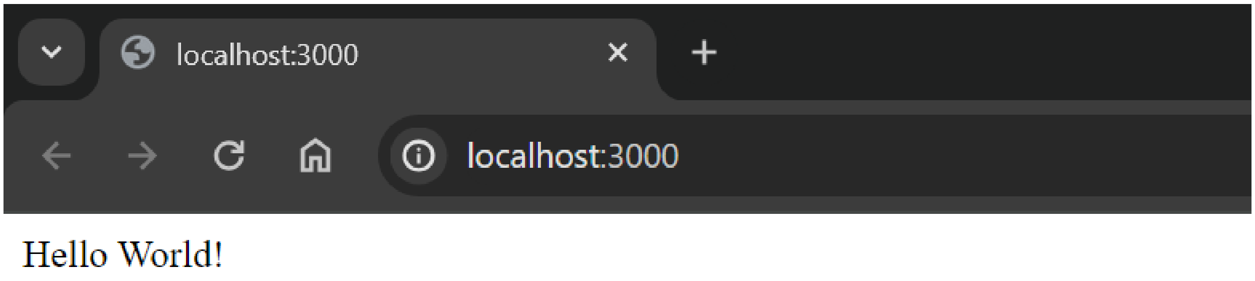
```
PS C:\Users\Sarif Iqbal\Downloads\webDev> node -v
v18.18.2
PS C:\Users\Sarif Iqbal\Downloads\webDev> npm -v
10.4.0
PS C:\Users\Sarif Iqbal\Downloads\webDev> npm init -y
Wrote to C:\Users\Sarif Iqbal\Downloads\webDev\package.json:

{
  "name": "webdev",
  "version": "1.0.0",
  "description": "",
  "main": "app.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}
```



```
PS C:\Users\Sarif Iqbal\Downloads\webDev> npm install express

added 64 packages, and audited 65 packages in 5s

12 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
PS C:\Users\Sarif Iqbal\Downloads\webDev>
```



```
PS C:\Users\Sarif Iqbal\Downloads\webDev> node app.js
Server is running on http://localhost:3000
```



Hello World!

Experiment – 6

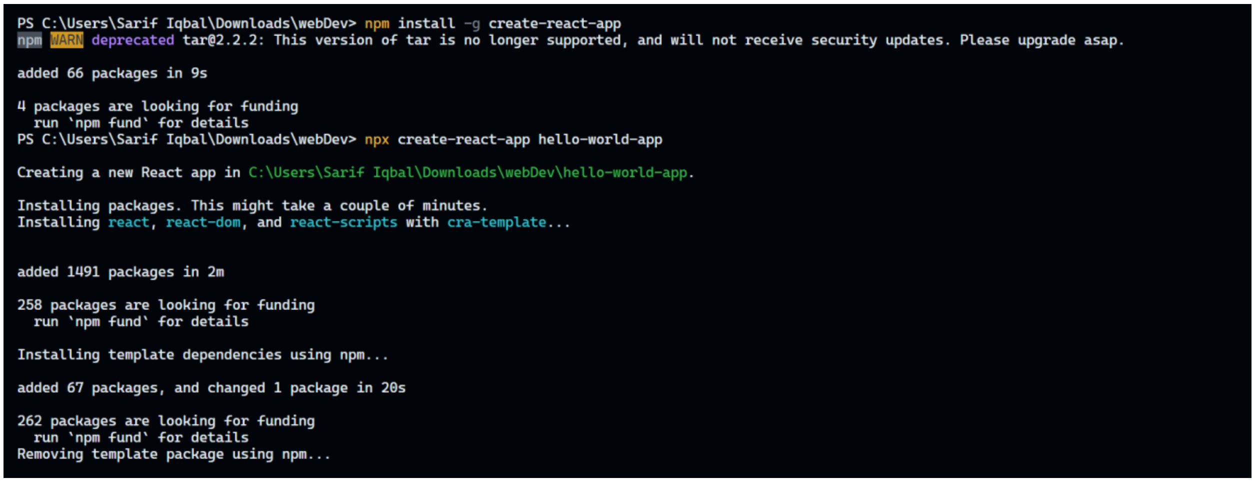
Aim: Set up React App and print “Hello World”

Code:

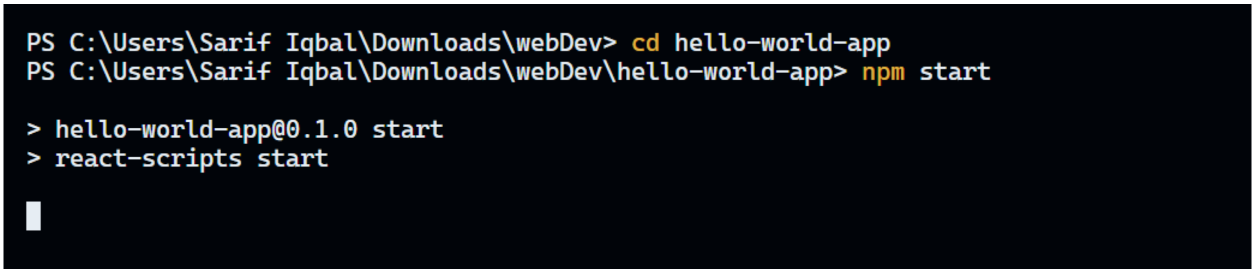
```
npm install -g create-react-app

npx create-react-app hello-world-app

cd hello-world-app

npm start
```

```
import React from 'react';

function App() {
  return (
    <div>
      <h1>Hello World</h1>
    </div>
  );
}

export default App;
```

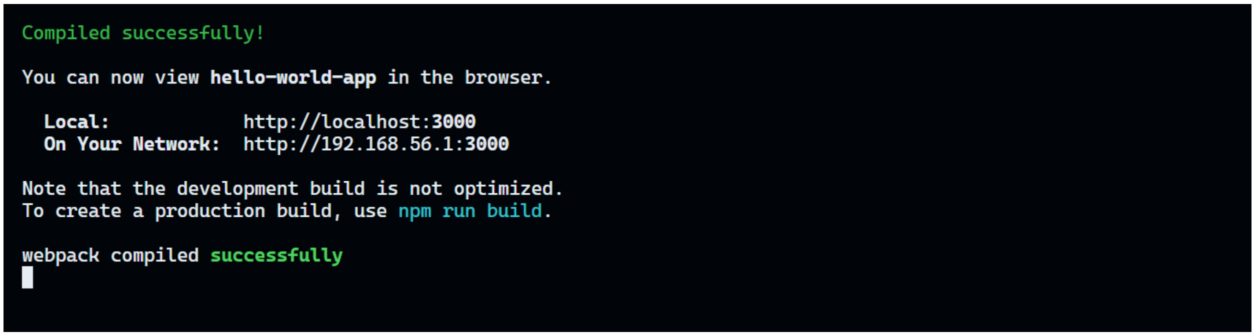
Output:



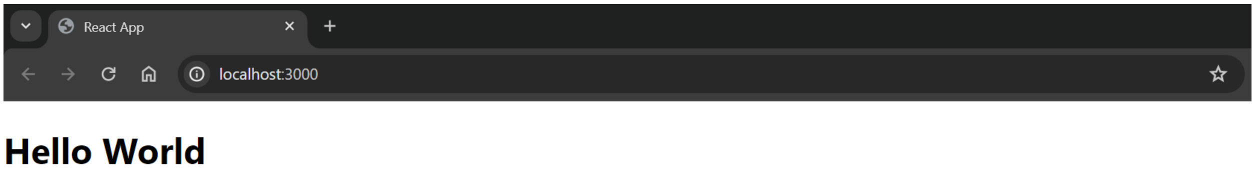
```
PS C:\Users\Sarif Iqbal\Downloads\webDev> npm install -g create-react-app
npm WARN deprecated tar@2.2.2: This version of tar is no longer supported, and will not receive security updates. Please upgrade asap.

added 66 packages in 9s

4 packages are looking for funding
  run `npm fund` for details
PS C:\Users\Sarif Iqbal\Downloads\webDev> npx create-react-app hello-world-app

Creating a new React app in C:\Users\Sarif Iqbal\Downloads\webDev\hello-world-app.

Installing packages. This might take a couple of minutes.
Installing react, react-dom, and react-scripts with cra-template...


added 1491 packages in 2m

258 packages are looking for funding
  run `npm fund` for details

Installing template dependencies using npm...

added 67 packages, and changed 1 package in 20s

262 packages are looking for funding
  run `npm fund` for details
Removing template package using npm...
```



```
PS C:\Users\Sarif Iqbal\Downloads\webDev> cd hello-world-app
PS C:\Users\Sarif Iqbal\Downloads\webDev\hello-world-app> npm start

> hello-world-app@0.1.0 start
> react-scripts start

```



```
Compiled successfully!

You can now view hello-world-app in the browser.

  Local:            http://localhost:3000
  On Your Network:  http://192.168.56.1:3000

Note that the development build is not optimized.
To create a production build, use npm run build.

webpack compiled successfully
```



Hello World

Experiment – 7

Aim: To make list components and table components

Code:

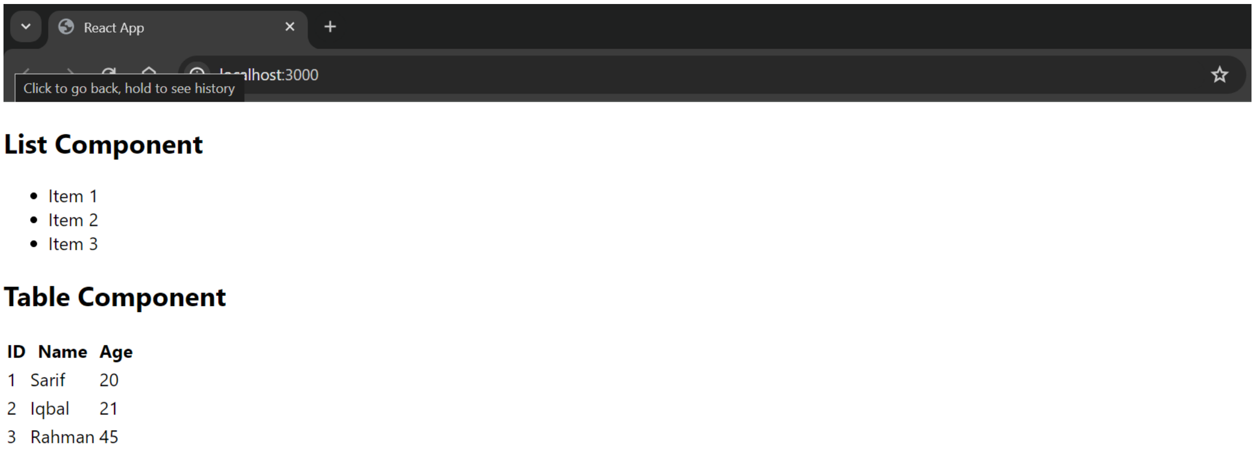
```
import React from 'react';

function List() {
  const items = ['Item 1', 'Item 2', 'Item 3'];

  return (
    <div>
      <h2>List Component</h2>
      <ul>
        {items.map((item, index) => (
          <li key={index}>{item}</li>
        ))}
      </ul>
    </div>
  );
}

export default List;
```

```
import React from 'react';

function Table() {
  const data = [
    { id: 1, name: 'Sarif', age: 20 },
    { id: 2, name: 'Iqbal', age: 21 },
    { id: 3, name: 'Rahman', age: 45 },
  ];

  return (
    <div>
      <h2>Table Component</h2>
      <table>
        <thead>
```

```
          <tr>
            <th>ID</th>
            <th>Name</th>
            <th>Age</th>
          </tr>
        </thead>
        <tbody>
          {data.map((item) => (
            <tr key={item.id}>
              <td>{item.id}</td>
              <td>{item.name}</td>
              <td>{item.age}</td>
            </tr>
          ))}
        </tbody>
      </table>
    </div>
  );
}

export default Table;
```

```
import React from 'react';
import List from './List';
import Table from './Table';

function App() {
  return (
    <div>
      <List />
      <Table />
    </div>
  );
}

export default App;
```

Output:



List Component

- Item 1
- Item 2
- Item 3

Table Component

ID	Name	Age
1	Sarif	20
2	Iqbal	21
3	Rahman	45

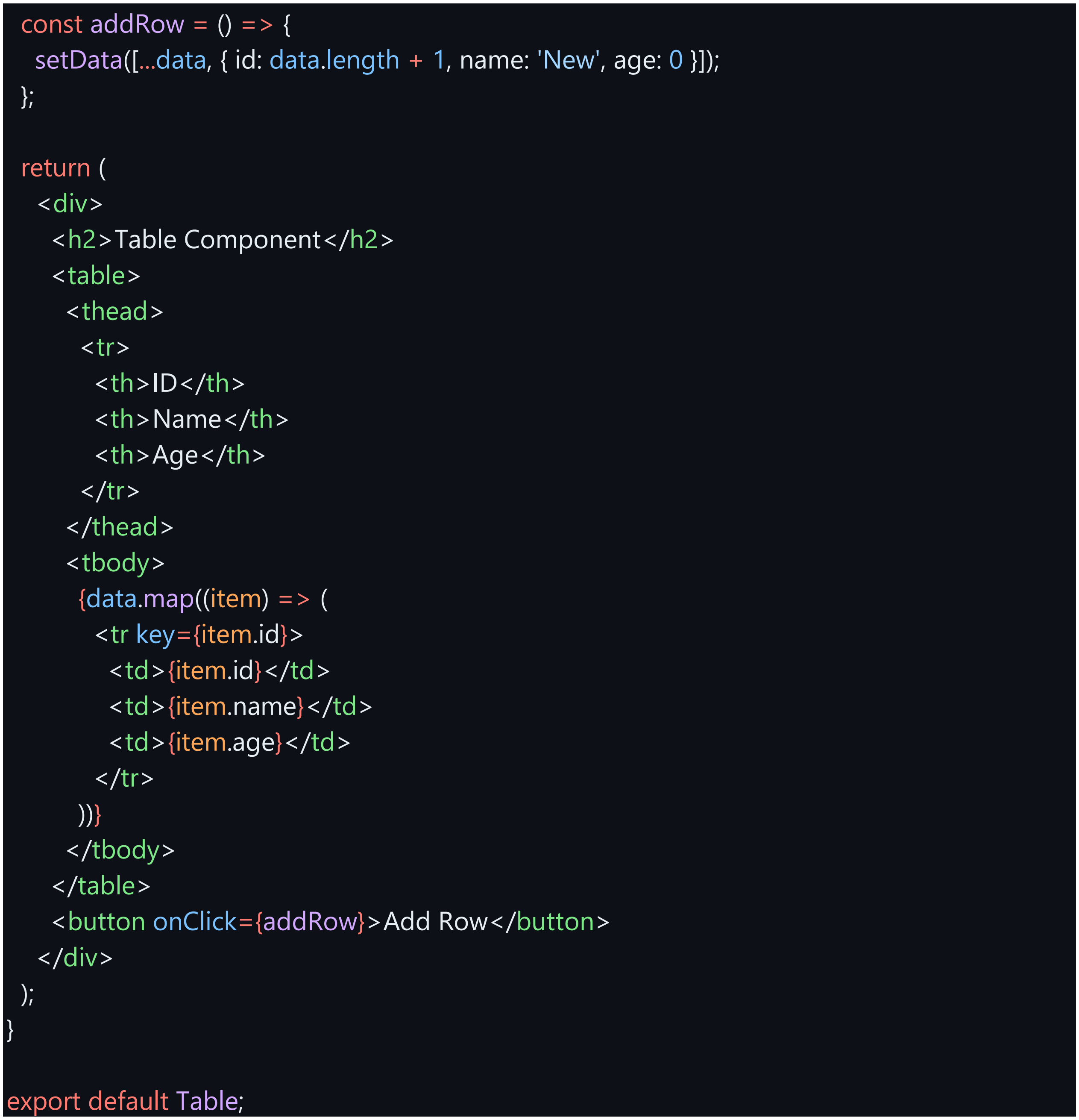
Experiment – 8 a

Aim: To perform use state in React to alter the state of components

Code:

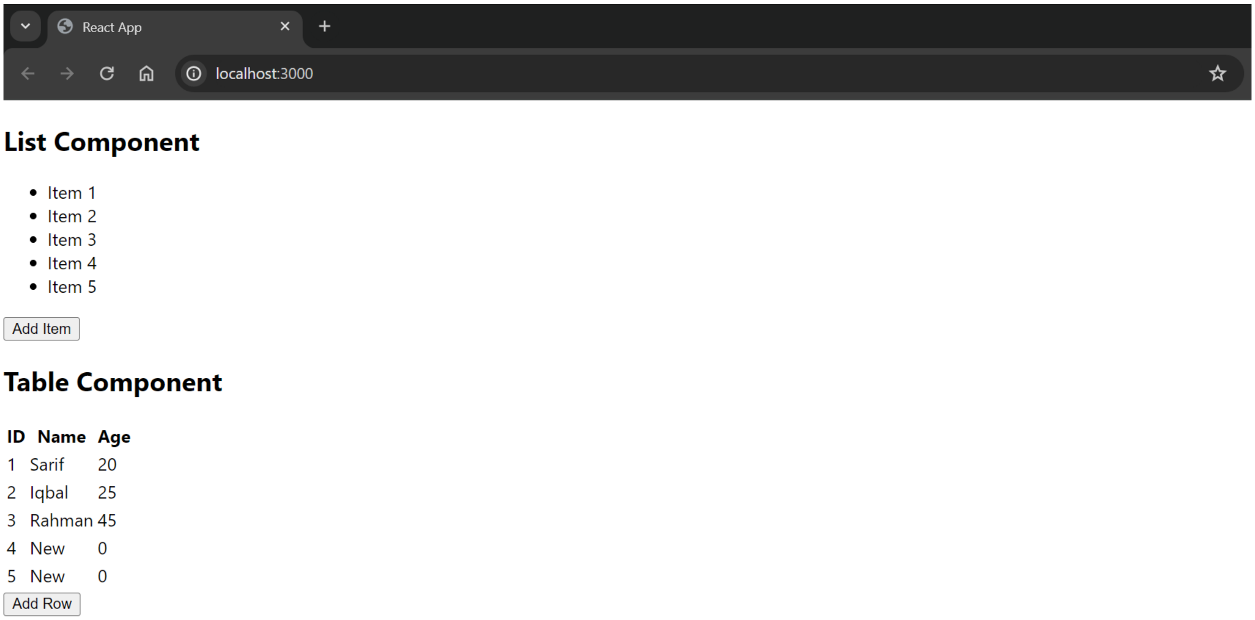
```
import React, { useState } from 'react';

function List() {
  const [items, setItems] = useState(['Item 1', 'Item 2', 'Item 3']);
  const addItem = () => {
    setItems([...items, `Item ${items.length + 1}`]);
  };

  return (
    <div>
      <h2>List Component</h2>
      <ul>
        {items.map((item, index) => (
          <li key={index}>{item}</li>
        ))}
      </ul>
      <button onClick={addItem}>Add Item</button>
    </div>
  );
}

export default List;
```

```
import React, { useState } from 'react';

function Table() {
  const [data, setData] = useState([
    { id: 1, name: 'Sarif', age: 20 },
    { id: 2, name: 'Iqbal', age: 25 },
    { id: 3, name: 'Rahman', age: 45 },
  ]);
```



```
  const addRow = () => {
    setData([...data, { id: data.length + 1, name: 'New', age: 0 }]);
  };

  return (
    <div>
      <h2>Table Component</h2>
      <table>
        <thead>
          <tr>
            <th>ID</th>
            <th>Name</th>
            <th>Age</th>
          </tr>
        </thead>
        <tbody>
          {data.map((item) => (
            <tr key={item.id}>
              <td>{item.id}</td>
              <td>{item.name}</td>
              <td>{item.age}</td>
            </tr>
          ))}
        </tbody>
      </table>
      <button onClick={addRow}>Add Row</button>
    </div>
  );
}

export default Table;
```

Output:



List Component

- Item 1
- Item 2
- Item 3
- Item 4
- Item 5

Add Item

Table Component

ID	Name	Age
1	Sarif	20
2	Iqbal	25
3	Rahman	45
4	New	0
5	New	0

Add Row

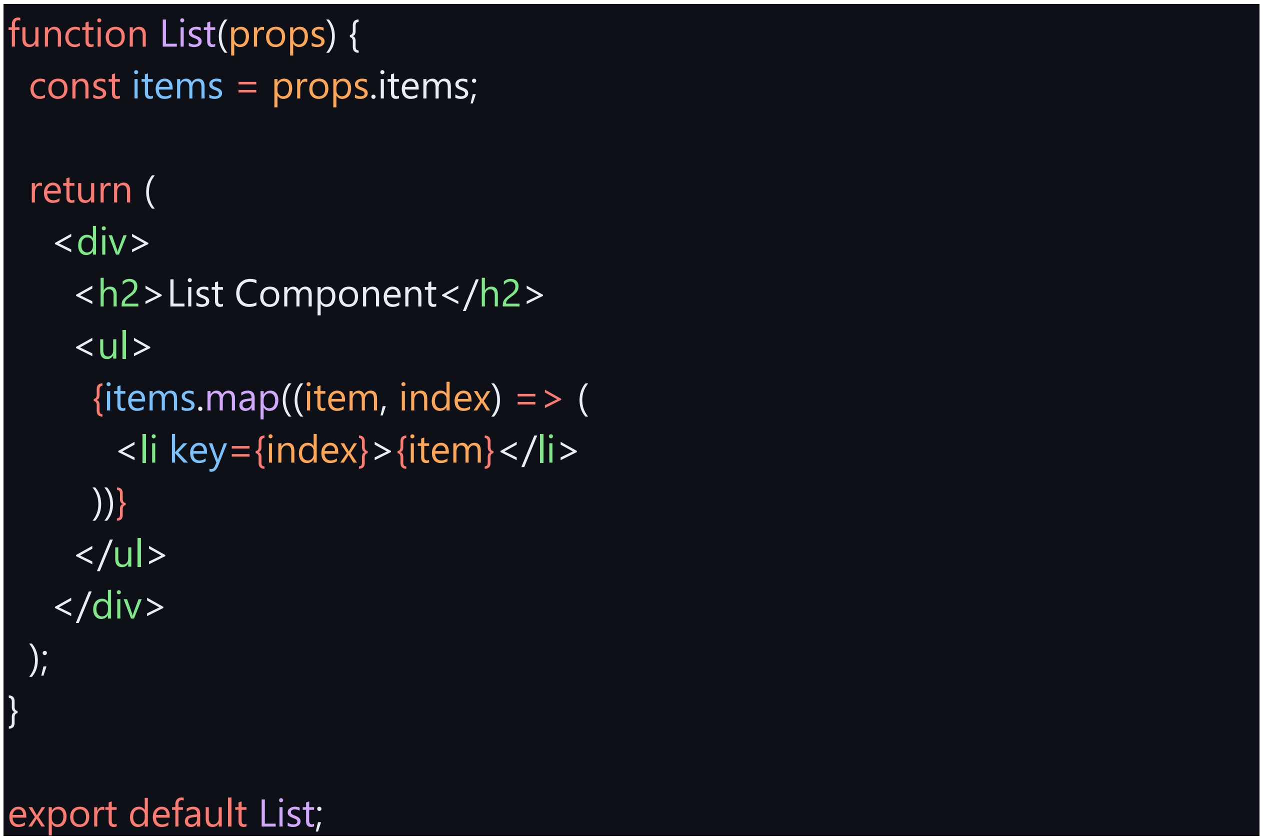
Experiment – 8 b

Aim: Use props to send data between components

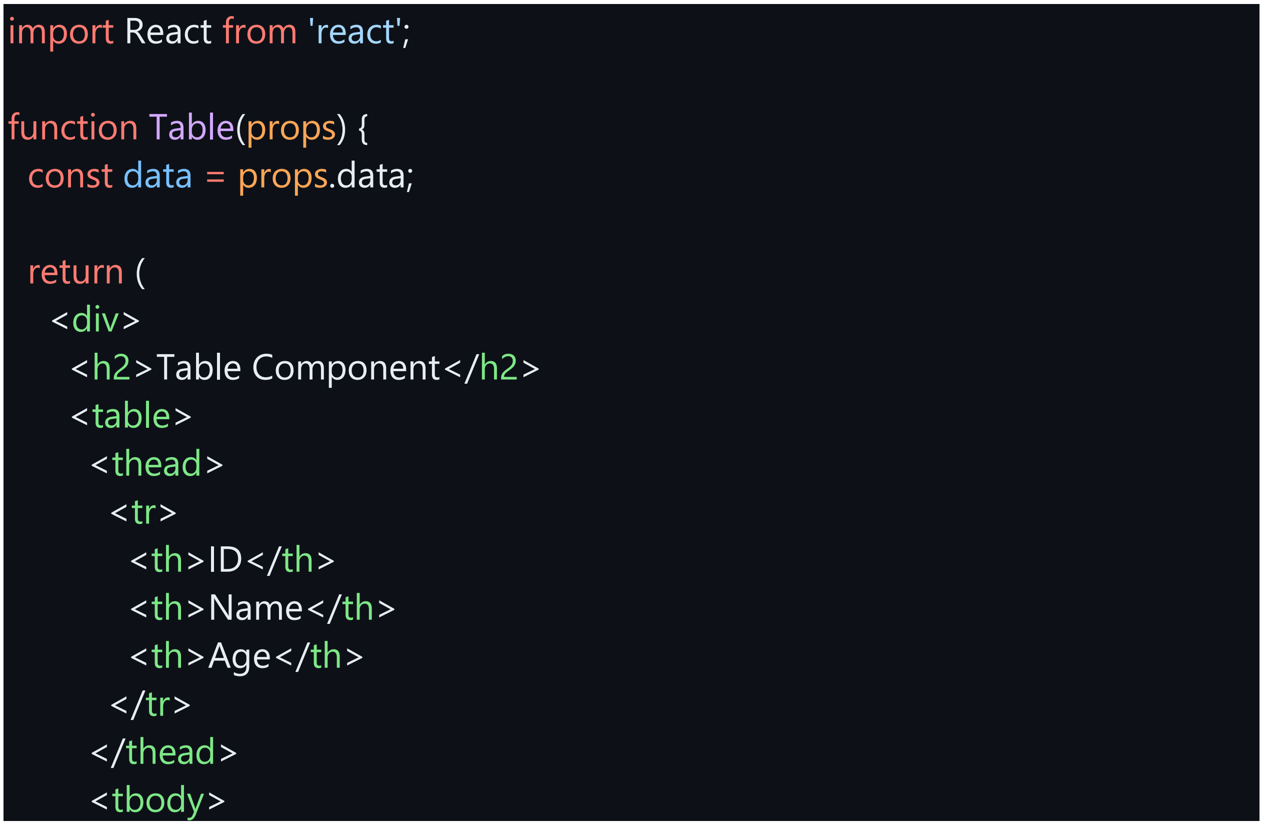
Code:

```
import React from 'react';
import List from './List';
import Table from './Table';

function App() {
  const listItems = ['Item 1', 'Item 2', 'Item 3'];

  const tableData = [
    { id: 1, name: 'Sarif', age: 20 },
    { id: 2, name: 'Iqbal', age: 21 },
    { id: 3, name: 'Rahman', age: 45 },
  ];

  return (
    <div>
      <List items={listItems} />
      <Table data={tableData} />
    </div>
  );
}

export default App;
```

```
import React from 'react';

```



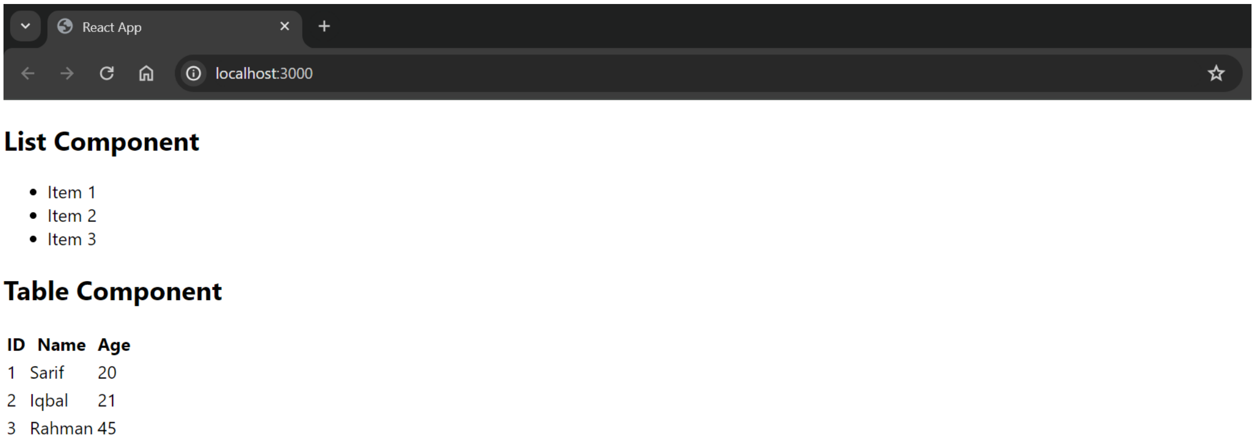
```
function List(props) {
  const items = props.items;

  return (
    <div>
      <h2>List Component</h2>
      <ul>
        {items.map((item, index) => (
          <li key={index}>{item}</li>
        ))}
      </ul>
    </div>
  );
}

export default List;
```



```
import React from 'react';

function Table(props) {
  const data = props.data;

  return (
    <div>
      <h2>Table Component</h2>
      <table>
        <thead>
          <tr>
            <th>ID</th>
            <th>Name</th>
            <th>Age</th>
          </tr>
        </thead>
        <tbody>
```

```
          {data.map((item) => (
            <tr key={item.id}>
              <td>{item.id}</td>
              <td>{item.name}</td>
              <td>{item.age}</td>
            </tr>
          ))}
        </tbody>
      </table>
    </div>
  );
}

export default Table;
```

Output:



List Component

- Item 1
- Item 2
- Item 3

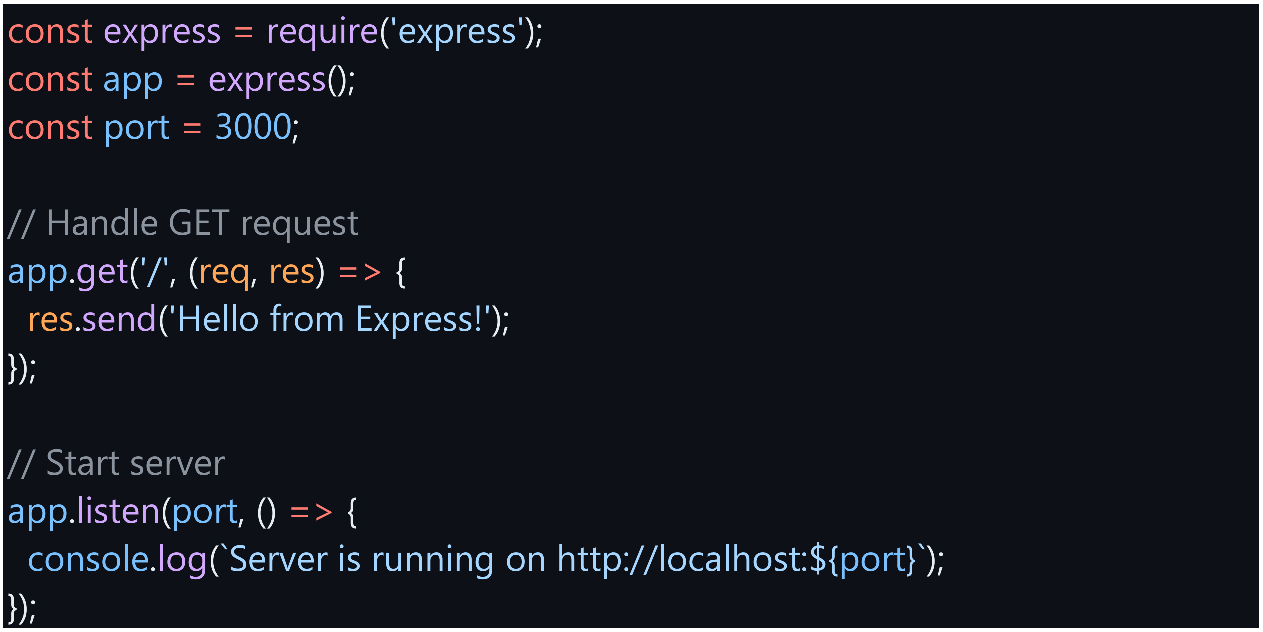
Table Component

ID	Name	Age
1	Sarif	20
2	Iqbal	21
3	Rahman	45

Experiment – 9

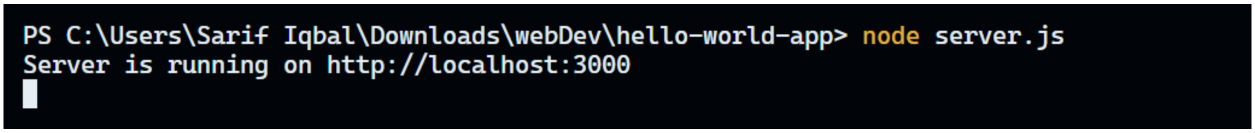
Aim: To create a server in Node.JS and Express.JS and send a get request

Code:

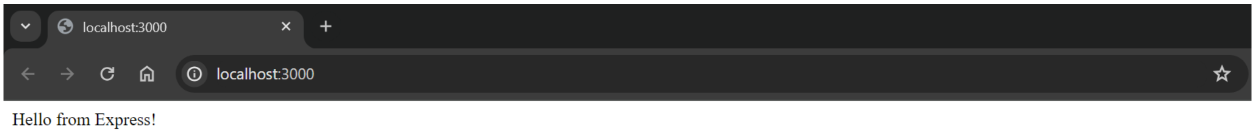


```
const express = require('express');
const app = express();
const port = 3000;

// Handle GET request
app.get('/', (req, res) => {
  res.send('Hello from Express!');
});

// Start server
app.listen(port, () => {
  console.log(`Server is running on http://localhost:${port}`);
});
```

Output:



```
PS C:\Users\Sarif Iqbal\Downloads\webDev\hello-world-app> node server.js
Server is running on http://localhost:3000
```



Hello from Express!

Experiment – 10 a

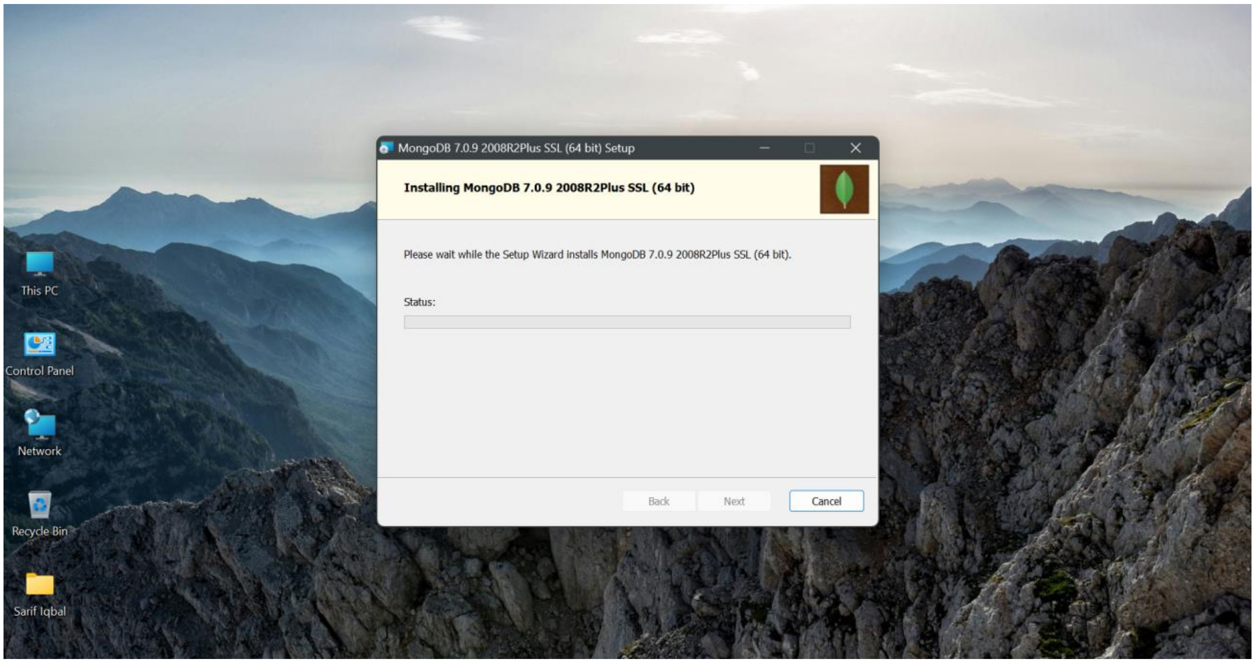
Aim: To install MongoDB Server and Mongosh on the local machine

Code:

```
mongod --version
```

```
mongosh --version
```

Output:

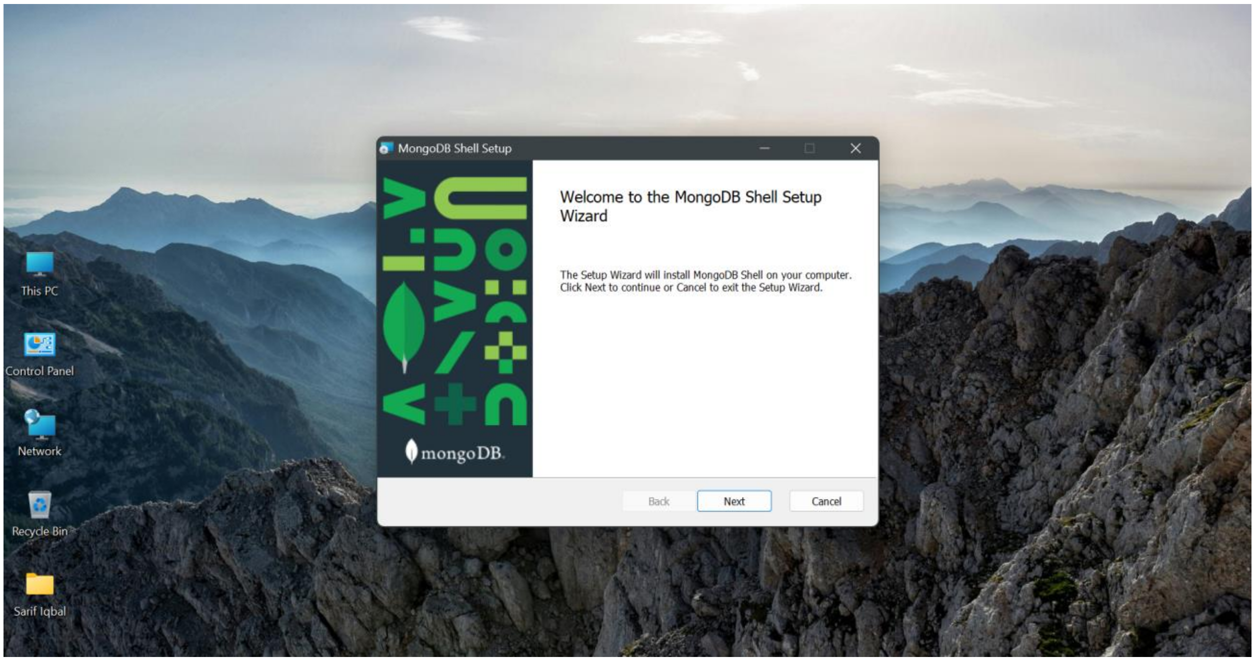


MongoDB 7.0.9 2008R2Plus SSL (64 bit) Setup

Installing MongoDB 7.0.9 2008R2Plus SSL (64 bit)

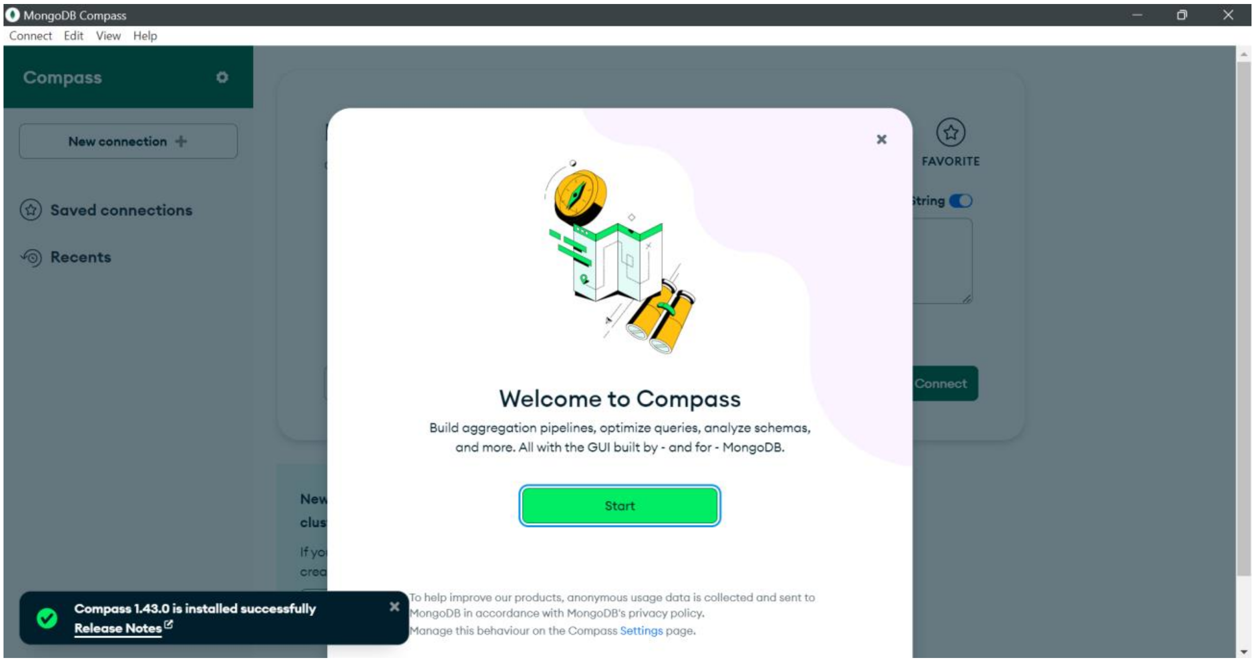
Please wait while the Setup Wizard installs MongoDB 7.0.9 2008R2Plus SSL (64 bit).

Status:



Welcome to the MongoDB Shell Setup Wizard

The Setup Wizard will install MongoDB Shell on your computer. Click Next to continue or Cancel to exit the Setup Wizard.

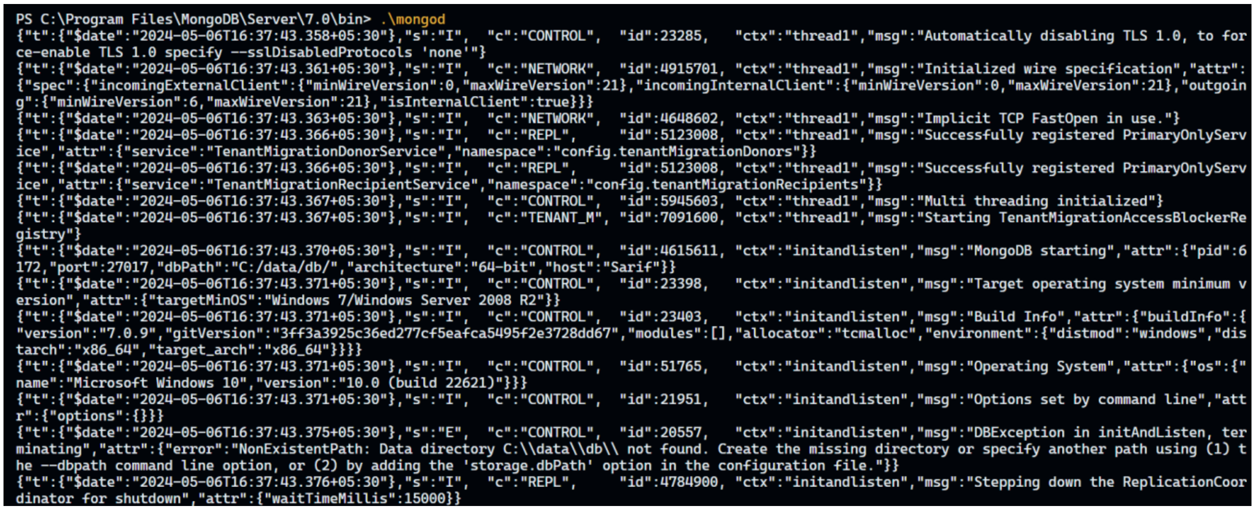


Welcome to Compass

Build aggregation pipelines, optimize queries, analyze schemas, and more. All with the GUI built by - and for - MongoDB.

To help improve our products, anonymous usage data is collected and sent to MongoDB in accordance with MongoDB's privacy policy.
Manage this behaviour on the Compass Settings page.

Compass 1.43.0 is installed successfully



```
PS C:\Program Files\MongoDB\Server\7.0\bin> .\mongod
{"t":{"$date":"2024-05-06T16:37:43.358+05:30"},"s":"I",  "c":"CONTROL",  "id":23285,   "ctx":"thread1","msg":"Automatically disabling TLS 1.0, to for
ce-enable TLS 1.0 specify --sslDisabledProtocols 'none'"}
{"t":{"$date":"2024-05-06T16:37:43.361+05:30"},"s":"I",  "c":"NETWORK",  "id":4915701, "ctx":"thread1","msg":"Initialized wire specification","attr":
{"spec":{"incomingExternalClient":{"minWireVersion":0,"maxWireVersion":21},"incomingInternalClient":{"minWireVersion":0,"maxWireVersion":21},"outgoin
g":{"minWireVersion":6,"maxWireVersion":21},"isInternalClient":true}}}
{"t":{"$date":"2024-05-06T16:37:43.363+05:30"},"s":"I",  "c":"NETWORK",  "id":4648602, "ctx":"thread1","msg":"Implicit TCP FastOpen in use."}
{"t":{"$date":"2024-05-06T16:37:43.366+05:30"},"s":"I",  "c":"REPL",     "id":5123008, "ctx":"thread1","msg":"Successfully registered PrimaryOnlyServ
ice","attr":{"service":"TenantMigrationDonorService","namespace":"config.tenantMigrationDonors"}}
{"t":{"$date":"2024-05-06T16:37:43.366+05:30"},"s":"I",  "c":"REPL",     "id":5123008, "ctx":"thread1","msg":"Successfully registered PrimaryOnlyServ
ice","attr":{"service":"TenantMigrationRecipientService","namespace":"config.tenantMigrationRecipients"}}
{"t":{"$date":"2024-05-06T16:37:43.367+05:30"},"s":"I",  "c":"CONTROL",  "id":5945603, "ctx":"thread1","msg":"Multi threading initialized"}
{"t":{"$date":"2024-05-06T16:37:43.367+05:30"},"s":"I",  "c":"TENANT_M", "id":7091600, "ctx":"thread1","msg":"Starting TenantMigrationAccessBlockerRe
gistry"}
{"t":{"$date":"2024-05-06T16:37:43.370+05:30"},"s":"I",  "c":"CONTROL",  "id":4615611, "ctx":"initandlisten","msg":"MongoDB starting","attr":{"pid":6
172,"port":27017,"dbPath":"C:/data/db/","architecture":"64-bit","host":"Sarif"}}
{"t":{"$date":"2024-05-06T16:37:43.371+05:30"},"s":"I",  "c":"CONTROL",  "id":23398,   "ctx":"initandlisten","msg":"Target operating system minimum v
ersion","attr":{"targetMinOS":"Windows 7/Windows Server 2008 R2"}}
{"t":{"$date":"2024-05-06T16:37:43.371+05:30"},"s":"I",  "c":"CONTROL",  "id":23403,   "ctx":"initandlisten","msg":"Build Info","attr":{"buildInfo":{
"version":"7.0.9","gitVersion":"3ff3a3925c36ed277cf5eafca5495f2e3728dd67","modules":[],"allocator":"tcmalloc","environment":{"distmod":"windows","dis
tarch":"x86_64","target_arch":"x86_64"}}}}
{"t":{"$date":"2024-05-06T16:37:43.371+05:30"},"s":"I",  "c":"CONTROL",  "id":51765,   "ctx":"initandlisten","msg":"Operating System","attr":{"os":{"
name":"Microsoft Windows 10","version":"10.0 (build 22621)"}}}
{"t":{"$date":"2024-05-06T16:37:43.371+05:30"},"s":"I",  "c":"CONTROL",  "id":21951,   "ctx":"initandlisten","msg":"Options set by command line","att
r":{"options":{}}}
{"t":{"$date":"2024-05-06T16:37:43.375+05:30"},"s":"E",  "c":"CONTROL",  "id":20557,   "ctx":"initandlisten","msg":"DBException in initAndListen, ter
minating","attr":{"error":"NonExistentPath: Data directory C:\\data\\db\\ not found. Create the missing directory or specify another path using (1) t
he --dbpath command line option, or (2) by adding the 'storage.dbPath' option in the configuration file."}}
{"t":{"$date":"2024-05-06T16:37:43.376+05:30"},"s":"I",  "c":"REPL",     "id":4784900, "ctx":"initandlisten","msg":"Stepping down the ReplicationCoor
dinator for shutdown","attr":{"waitTimeMillis":15000}}
```

Experiment – 10 b

Aim: Create a MongoDB and perform CRUD operations on it

Code:

```
mongod
mongo
```

```
db.users.insertOne({ name: "Rahman", age: 25 });

db.users.insertMany([
  { name: "Sarif", age: 20 },
  { name: "Iqbal", age: 21 }
]);
```

```
db.users.find();

db.users.find({ age: { $gt: 25 } });
```

```
db.users.updateOne({ name: "Rahman" }, { $set: { age: 32 } });

db.users.updateMany({ age: { $gt: 30 } }, { $set: { status: "active" } });
```

```
db.users.deleteOne({ name: "Rahman" });

db.users.deleteMany({ status: "inactive" });
```

```
exit
```

Experiment – 11

Aim: Task management tool: Login/Register to the application, add daily tasks, Assign a due date of completion, Mark them as complete/incomplete, and View weekly/monthly statistics of their to-dos.

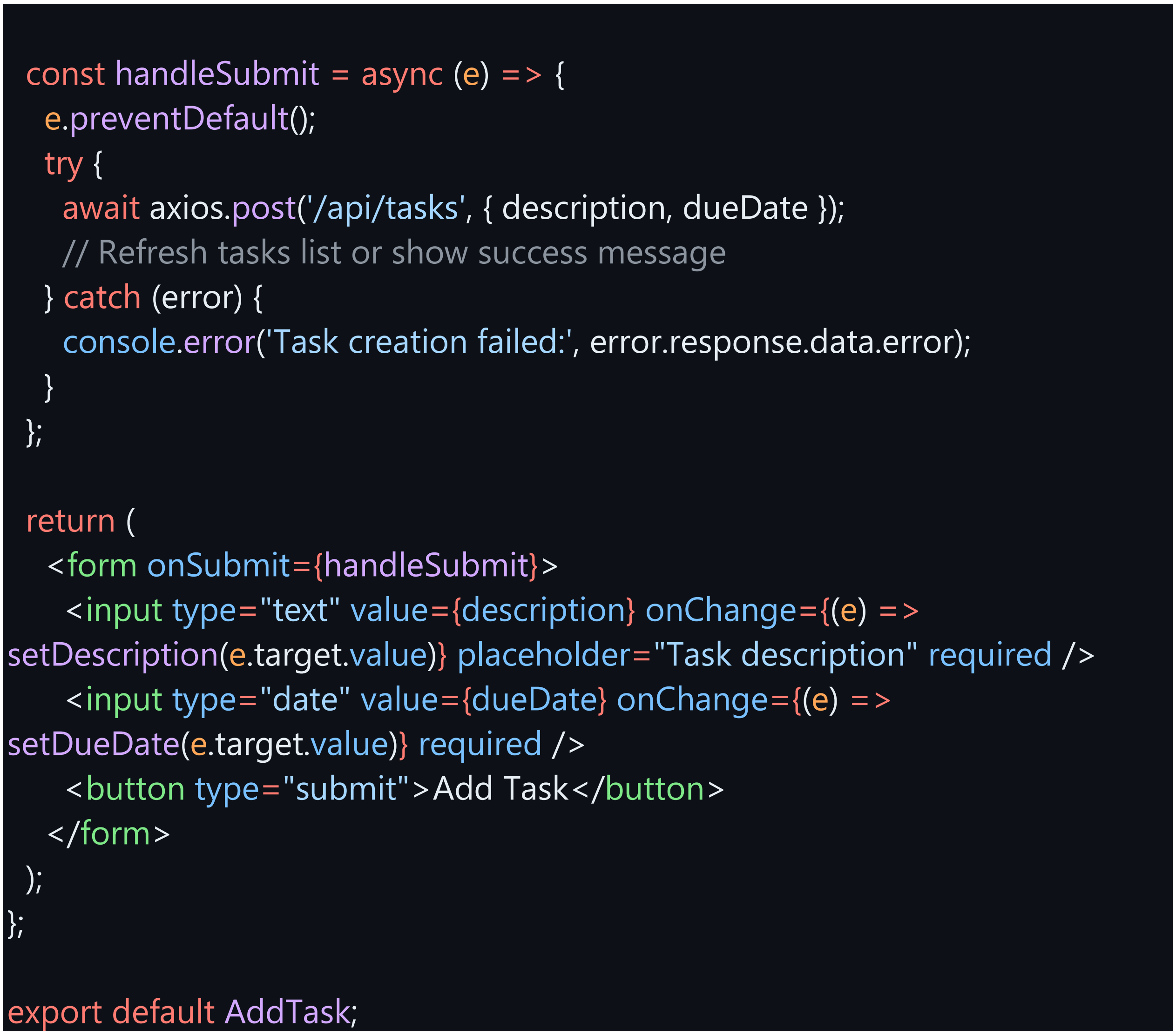
Code:

```
// server.js
const express = require('express');
const mongoose = require('mongoose');
const bodyParser = require('body-parser');
const cors = require('cors');

const app = express();

// Middleware
app.use(bodyParser.json());
app.use(cors());

// Connect to MongoDB
mongoose.connect('mongodb://localhost/taskmanager', {
  useNewUrlParser: true,
  useUnifiedTopology: true
})
.then(() => console.log('MongoDB connected'))
.catch(err => console.log(err));

// Routes
const authRoutes = require('./routes/auth');
const taskRoutes = require('./routes/tasks');

app.use('/api/auth', authRoutes);
app.use('/api/tasks', taskRoutes);

// Start the server
const PORT = process.env.PORT || 5000;
```

```
app.listen(PORT, () => console.log(`Server started on port ${PORT}`));
```

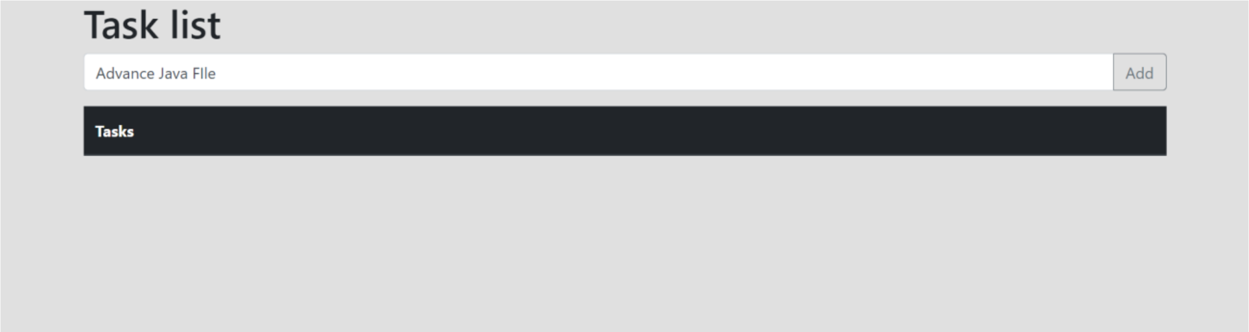
```
// models/Task.js
const mongoose = require('mongoose');

const taskSchema = new mongoose.Schema({
  description: {
    type: String,
    required: true
  },
  dueDate: {
    type: Date,
    required: true
  },
  completed: {
    type: Boolean,
    default: false
  },
  userId: {
    type: mongoose.Schema.Types.ObjectId,
    ref: 'User',
    required: true
  }
});

module.exports = mongoose.model('Task', taskSchema);
```

```
// components/Tasks/AddTask.js
import React, { useState } from 'react';
import axios from 'axios';

const AddTask = () => {
  const [description, setDescription] = useState('');
  const [dueDate, setDueDate] = useState('');
```



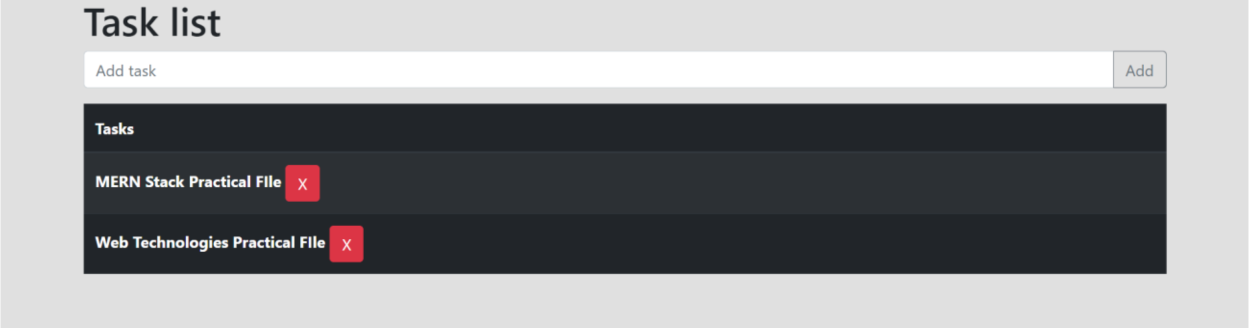
```
  const handleSubmit = async (e) => {
    e.preventDefault();
    try {
      await axios.post('/api/tasks', { description, dueDate });
      // Refresh tasks list or show success message
    } catch (error) {
      console.error('Task creation failed:', error.response.data.error);
    }
  };

  return (
    <form onSubmit={handleSubmit}>
      <input type="text" value={description} onChange={(e) => setDescription(e.target.value)} placeholder="Task description" required />
      <input type="date" value={dueDate} onChange={(e) => setDueDate(e.target.value)} required />
      <button type="submit">Add Task</button>
    </form>
  );
};

export default AddTask;
```

Output:



Task list

Advance Java FIle



Task list

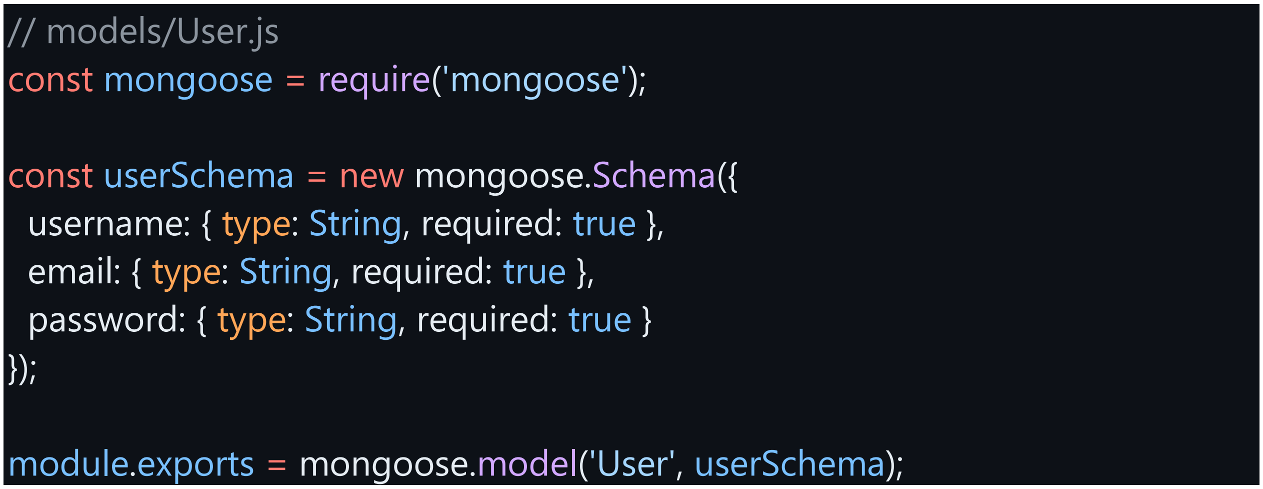
MERN Stack Practical FIle

Web Technologies Practical FIle

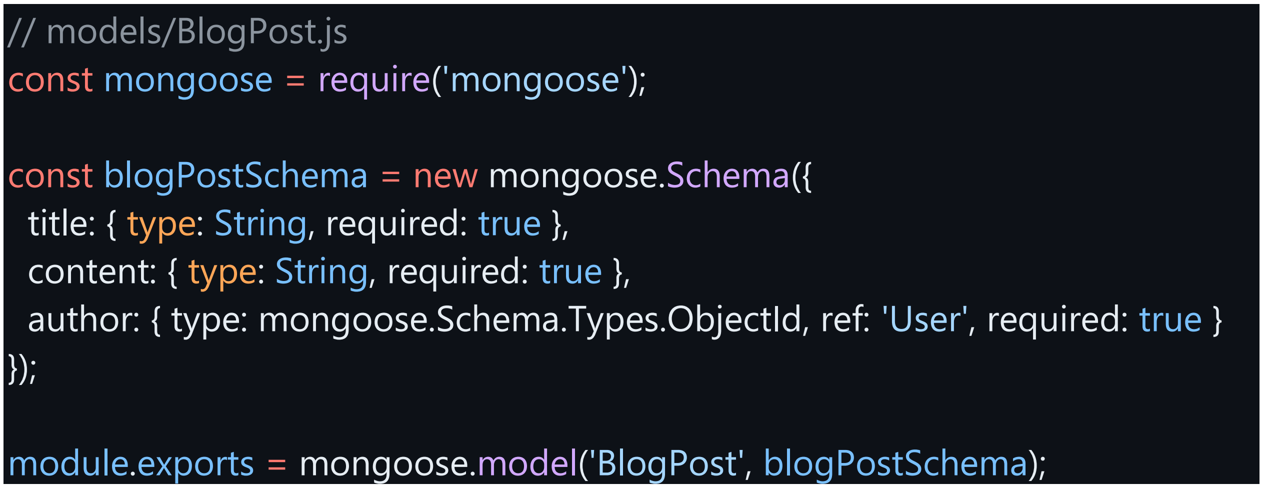
Experiment – 12

Aim: Blogging platform

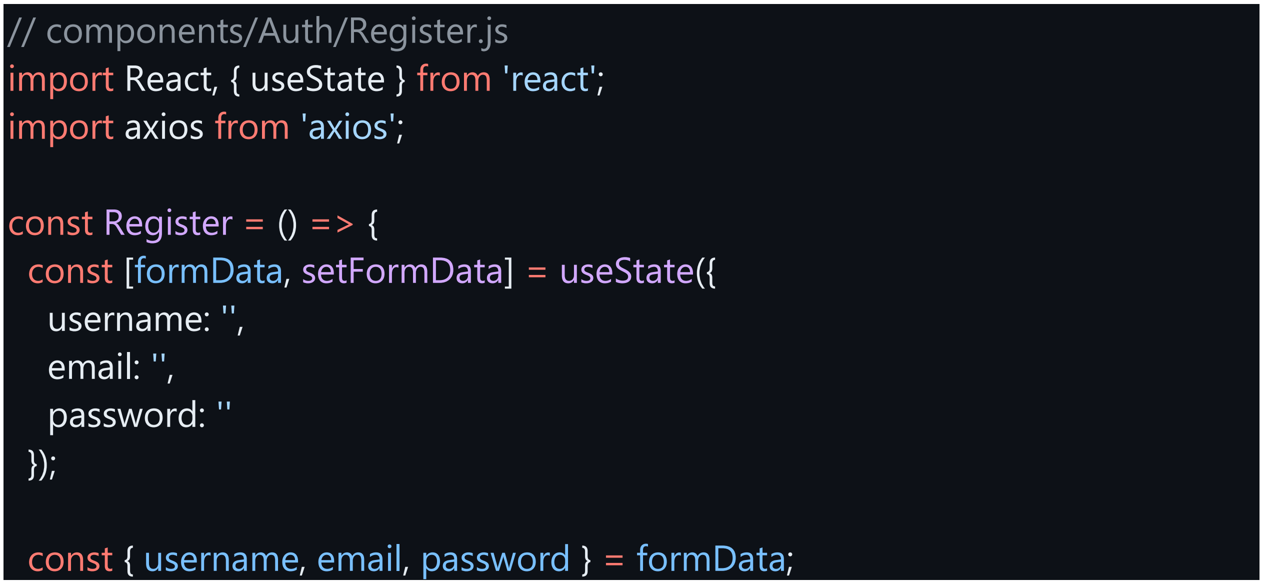
Code:



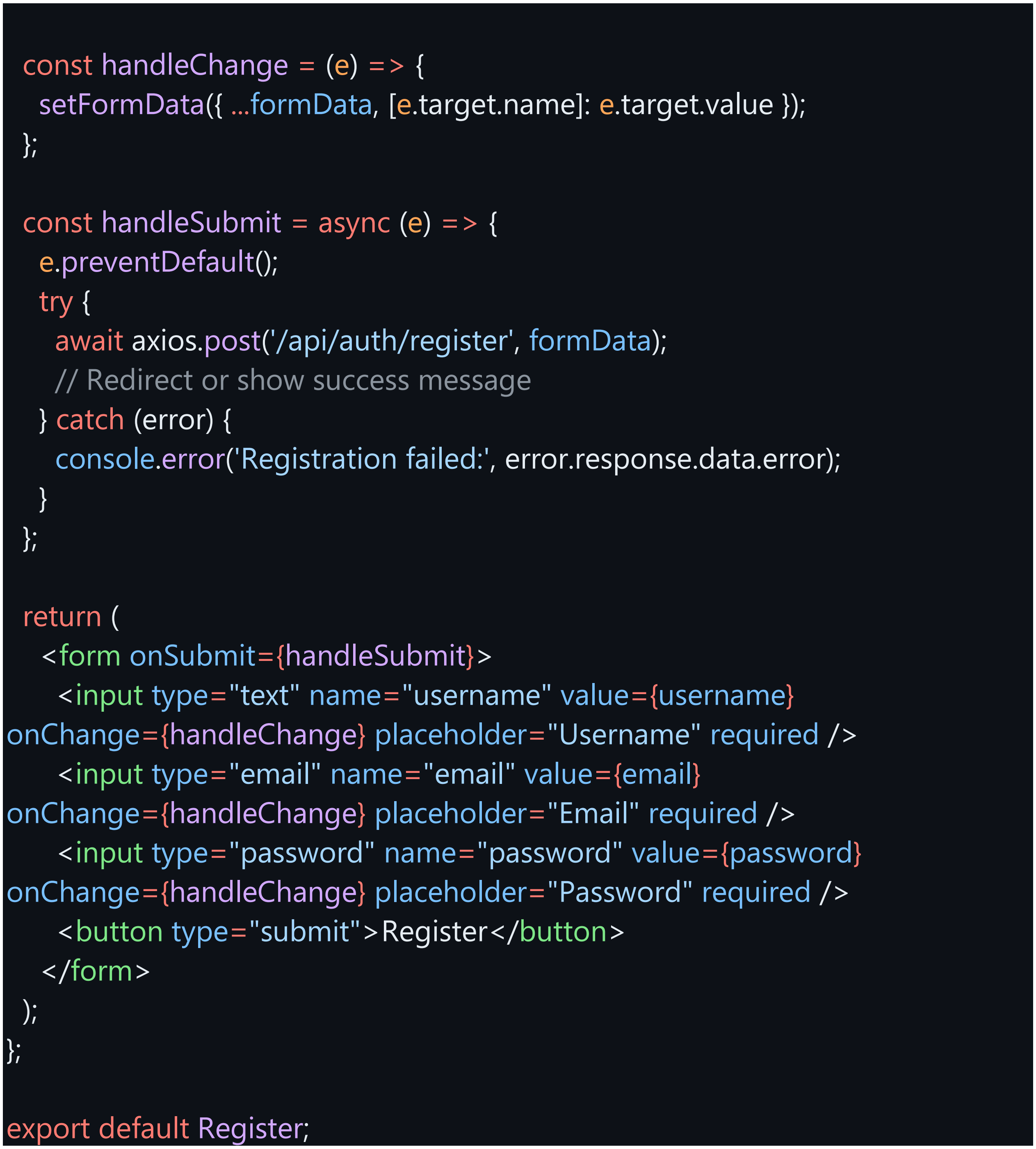
```
// models/User.js
const mongoose = require('mongoose');

const userSchema = new mongoose.Schema({
  username: { type: String, required: true },
  email: { type: String, required: true },
  password: { type: String, required: true }
});

module.exports = mongoose.model('User', userSchema);
```



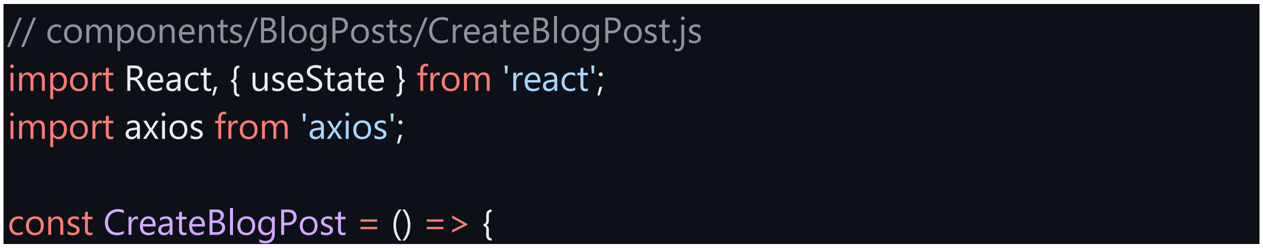
```
// models/BlogPost.js
const mongoose = require('mongoose');

const blogPostSchema = new mongoose.Schema({
  title: { type: String, required: true },
  content: { type: String, required: true },
  author: { type: mongoose.Schema.Types.ObjectId, ref: 'User', required: true }
});

module.exports = mongoose.model('BlogPost', blogPostSchema);
```



```
// components/Auth/Register.js
import React, { useState } from 'react';
import axios from 'axios';

const Register = () => {
  const [formData, setFormData] = useState({
    username: '',
    email: '',
    password: ''
  });

  const { username, email, password } = formData;
```

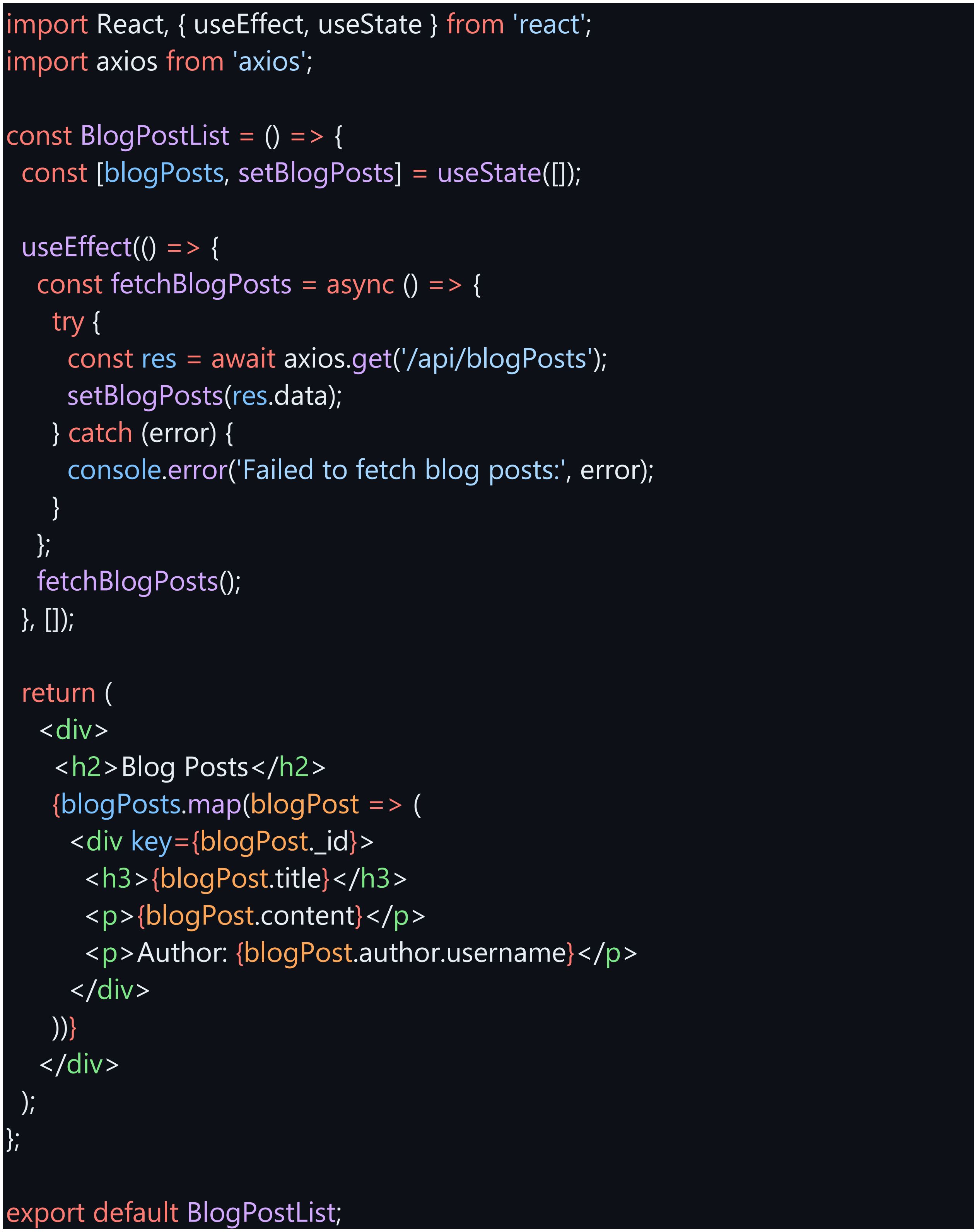


```
  const handleChange = (e) => {
    setFormData({ ...formData, [e.target.name]: e.target.value });
  };

  const handleSubmit = async (e) => {
    e.preventDefault();
    try {
      await axios.post('/api/auth/register', formData);
      // Redirect or show success message
    } catch (error) {
      console.error('Registration failed:', error.response.data.error);
    }
  };

  return (
    <form onSubmit={handleSubmit}>
      <input type="text" name="username" value={username}
onChange={handleChange} placeholder="Username" required />
      <input type="email" name="email" value={email}
onChange={handleChange} placeholder="Email" required />
      <input type="password" name="password" value={password}
onChange={handleChange} placeholder="Password" required />
      <button type="submit">Register</button>
    </form>
  );
};

export default Register;
```



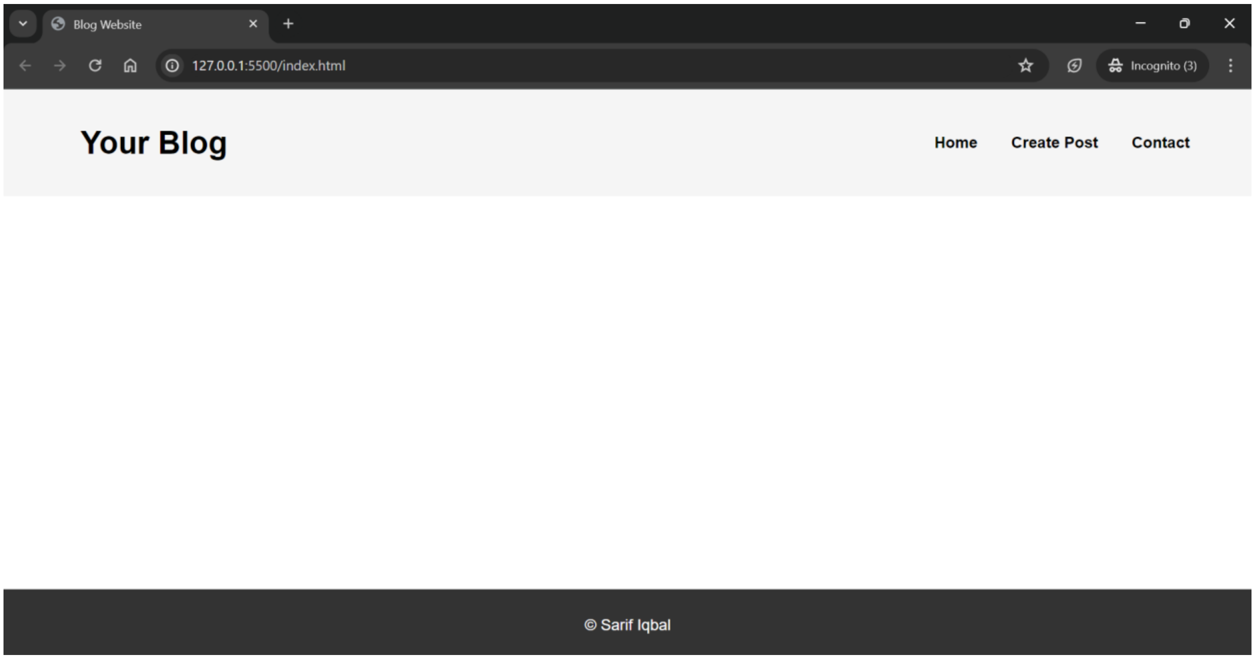
```
// components/BlogPosts/CreateBlogPost.js
import React, { useState } from 'react';
import axios from 'axios';

const CreateBlogPost = () => {
  const [formData, setFormData] = useState({
    title: '',
    content: ''
  });

  const { title, content } = formData;

  const handleChange = (e) => {
    setFormData({ ...formData, [e.target.name]: e.target.value });
  };

  const handleSubmit = async (e) => {
    e.preventDefault();
    try {
      await axios.post('/api/blogPosts', formData);
      // Redirect or show success message
    } catch (error) {
      console.error('Blog post creation failed:', error.response.data.error);
    }
  };

  return (
    <form onSubmit={handleSubmit}>
      <input type="text" name="title" value={title} onChange={handleChange}
placeholder="Title" required />
      <textarea name="content" value={content} onChange={handleChange}
placeholder="Content" required />
      <button type="submit">Create Blog Post</button>
    </form>
  );
};

export default CreateBlogPost;
```

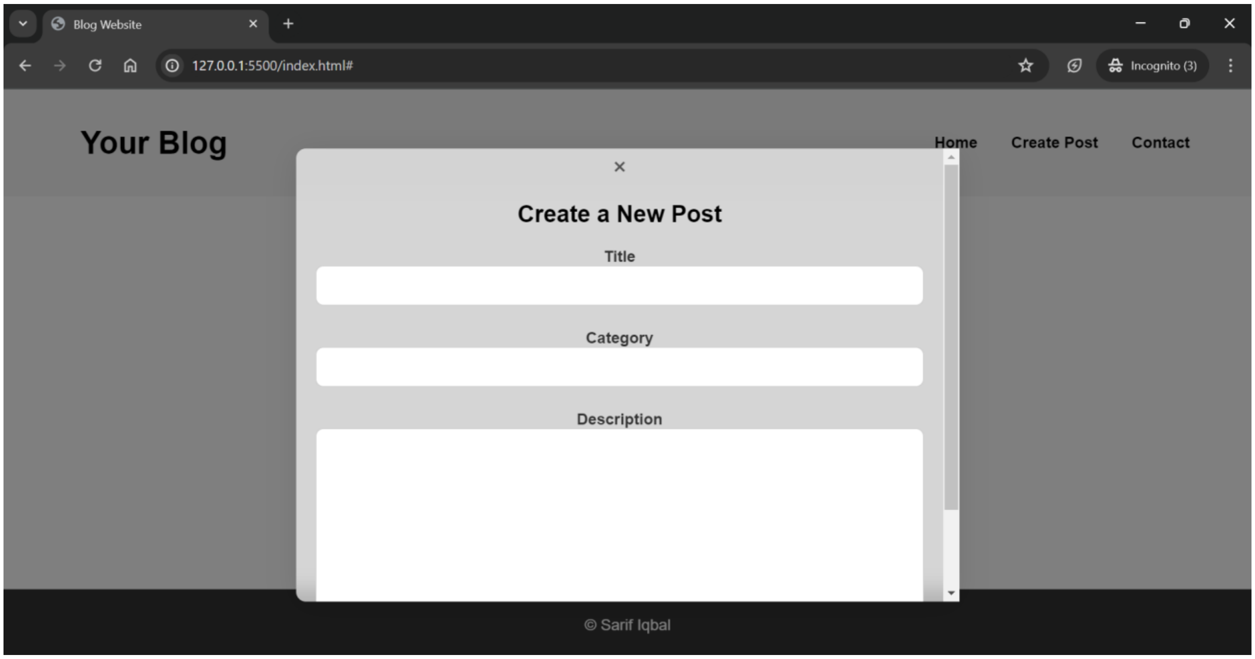
```
// components/BlogPosts/BlogPostList.js
```



```
import React, { useEffect, useState } from 'react';
import axios from 'axios';

const BlogPostList = () => {
  const [blogPosts, setBlogPosts] = useState([]);

  useEffect(() => {
    const fetchBlogPosts = async () => {
      try {
        const res = await axios.get('/api/blogPosts');
        setBlogPosts(res.data);
      } catch (error) {
        console.error('Failed to fetch blog posts:', error);
      }
    };
    fetchBlogPosts();
  }, []);

  return (
    <div>
      <h2>Blog Posts</h2>
      {blogPosts.map(blogPost => (
        <div key={blogPost._id}>
          <h3>{blogPost.title}</h3>
          <p>{blogPost.content}</p>
          <p>Author: {blogPost.author.username}</p>
        </div>
      ))}
    </div>
  );
};

export default BlogPostList;
```

Output:



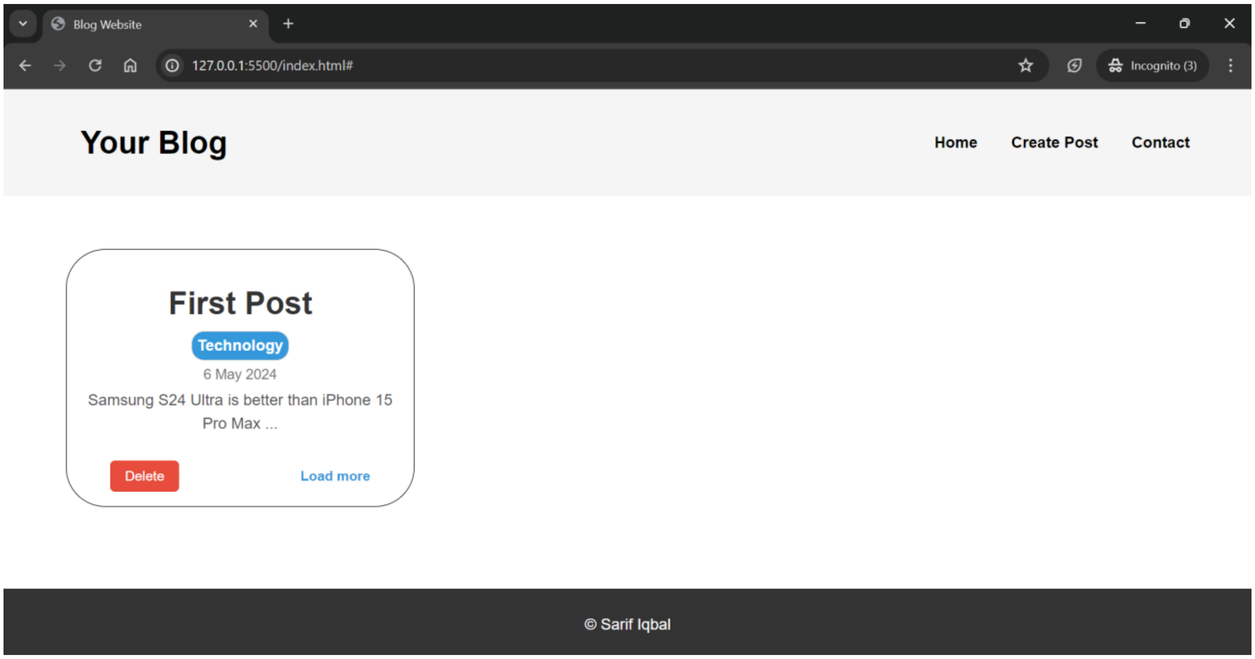


Create a New Post

Title

Category

Description



Your Blog

First Post

Technology

6 May 2024

Samsung S24 Ultra is better than iPhone 15 Pro Max ...

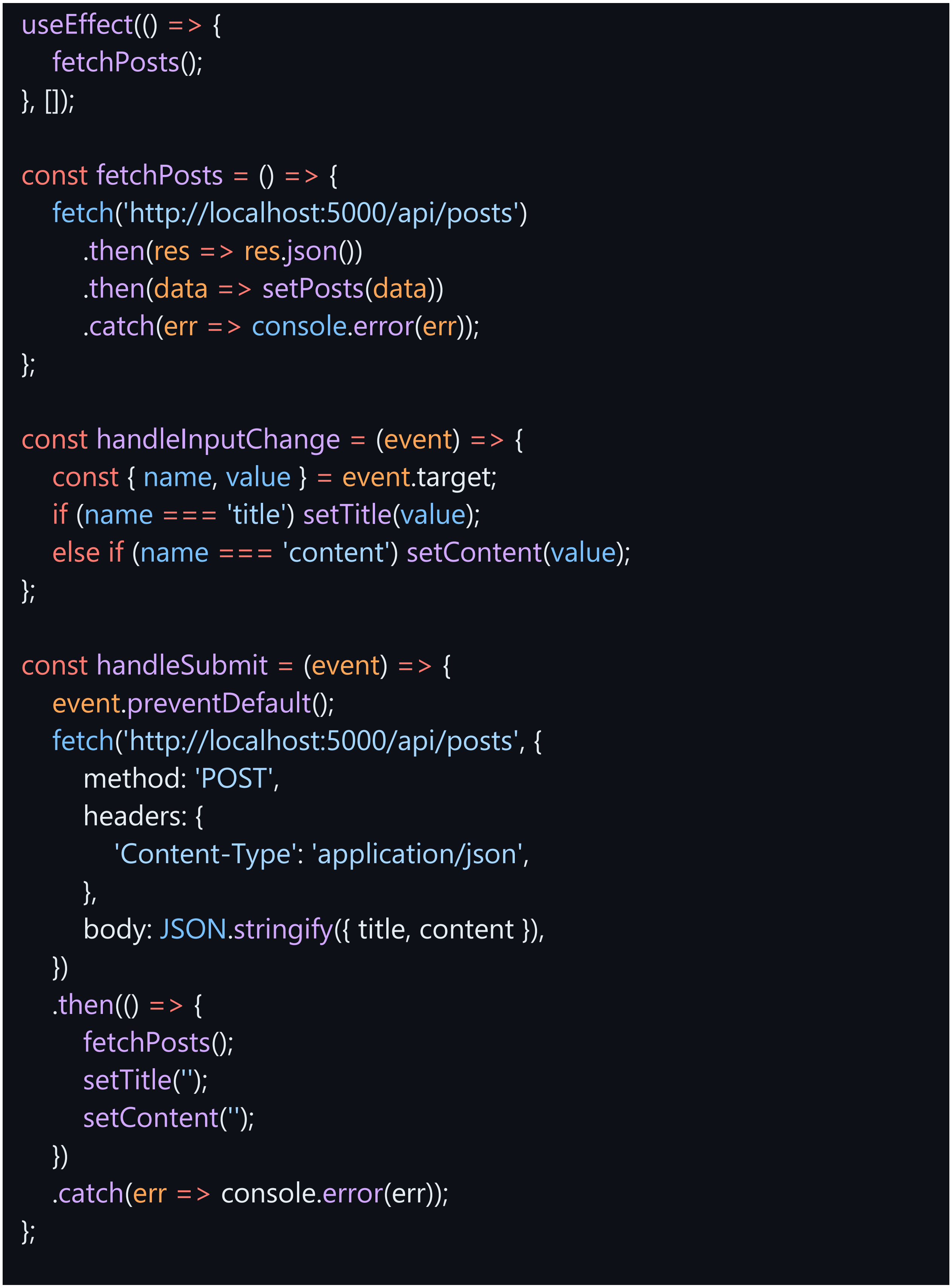
© Sarif Iqbal

Experiment – 13

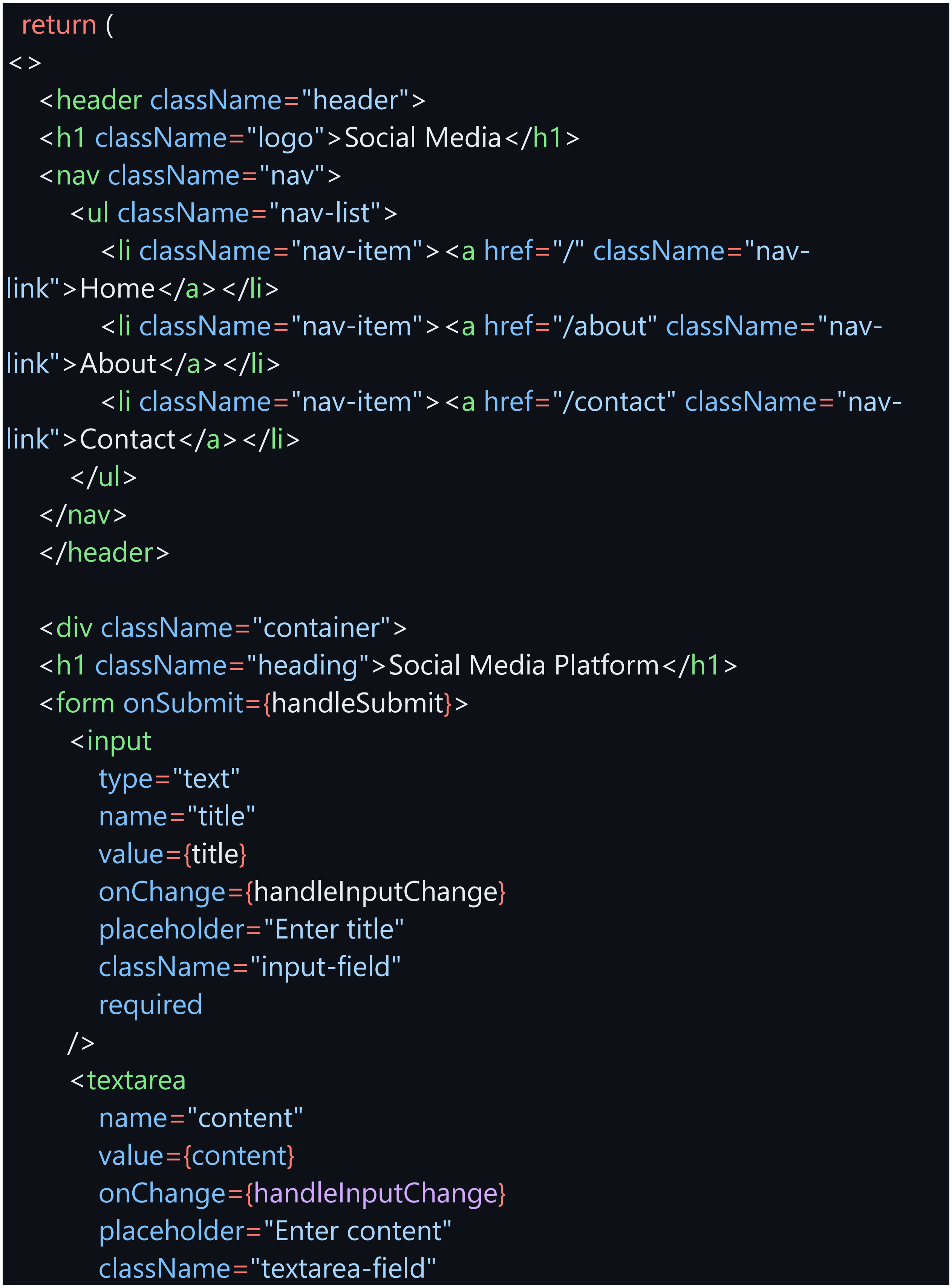
Aim: Social media platform

Code:

```
import React, { useState, useEffect } from 'react';

function App() {
  const [posts, setPosts] = useState([]);
  const [title, setTitle] = useState('');
  const [content, setContent] = useState('');
```



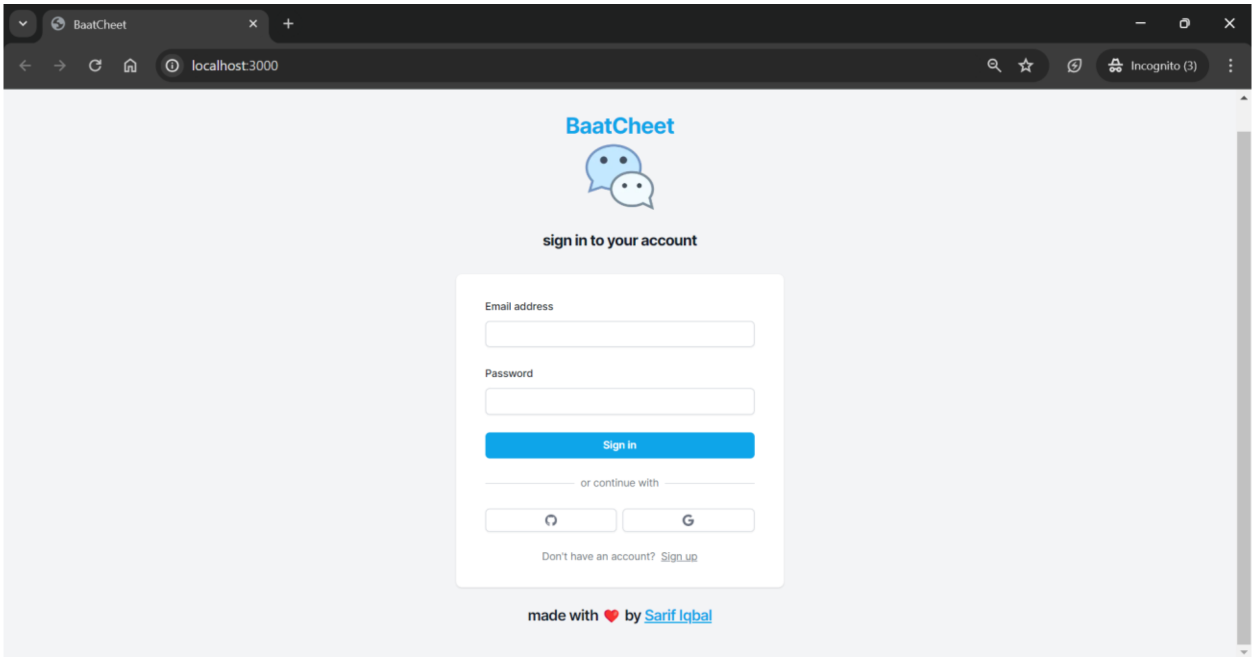
```
useEffect(() => {
  fetchPosts();
}, []);

const fetchPosts = () => {
  fetch('http://localhost:5000/api/posts')
    .then(res => res.json())
    .then(data => setPosts(data))
    .catch(err => console.error(err));
};

const handleInputChange = (event) => {
  const { name, value } = event.target;
  if (name === 'title') setTitle(value);
  else if (name === 'content') setContent(value);
};

const handleSubmit = (event) => {
  event.preventDefault();
  fetch('http://localhost:5000/api/posts', {
    method: 'POST',
    headers: {
      'Content-Type': 'application/json',
    },
    body: JSON.stringify({ title, content }),
  })
  .then(() => {
    fetchPosts();
    setTitle('');
    setContent('');
  })
  .catch(err => console.error(err));
};
```



```
 return (
<>
  <header className="header">
  <h1 className="logo">Social Media</h1>
  <nav className="nav">
    <ul className="nav-list">
      <li className="nav-item"><a href="/" className="nav-
link">Home</a></li>
      <li className="nav-item"><a href="/about" className="nav-
link">About</a></li>
      <li className="nav-item"><a href="/contact" className="nav-
link">Contact</a></li>
    </ul>
  </nav>
  </header>

  <div className="container">
  <h1 className="heading">Social Media Platform</h1>
  <form onSubmit={handleSubmit}>
    <input
      type="text"
      name="title"
      value={title}
      onChange={handleInputChange}
      placeholder="Enter title"
      className="input-field"
      required
    />
    <textarea
      name="content"
      value={content}
      onChange={handleInputChange}
      placeholder="Enter content"
      className="textarea-field"
```

```
          required
        ></textarea>
         <input
          type="file"
          accept="image/*"
          className="image-input"
         />
        <button type="submit" className="submit-button">Add
Post</button>
    </form>
    <ul className="post-list">
        {posts.map(post => (
          <li key={post._id} className="post-item">
            <img className='post-image'
src="https://images.unsplash.com/photo-1575936123452-
b67c3203c357?q=80&w=1000&auto=format&fit=crop&ixlib=rb-
4.0.3&ixid=M3wxMjA3fDB8MHxzZWFyY2h8Mnx8aW1hZ2V8ZW58MHx8MH
x8fDA%3D" alt="Image Description" />
              <h2 className="post-title">{post.title}</h2>
              <p className="post-content">{post.content}</p>
          </li>
        ))}
    </ul>
</div>

<footer className="footer">
          <p className="footer-text">© 2024 Social Media. All rights
reserved.</p>
        </footer>
</>
  );
}

export default App;
```

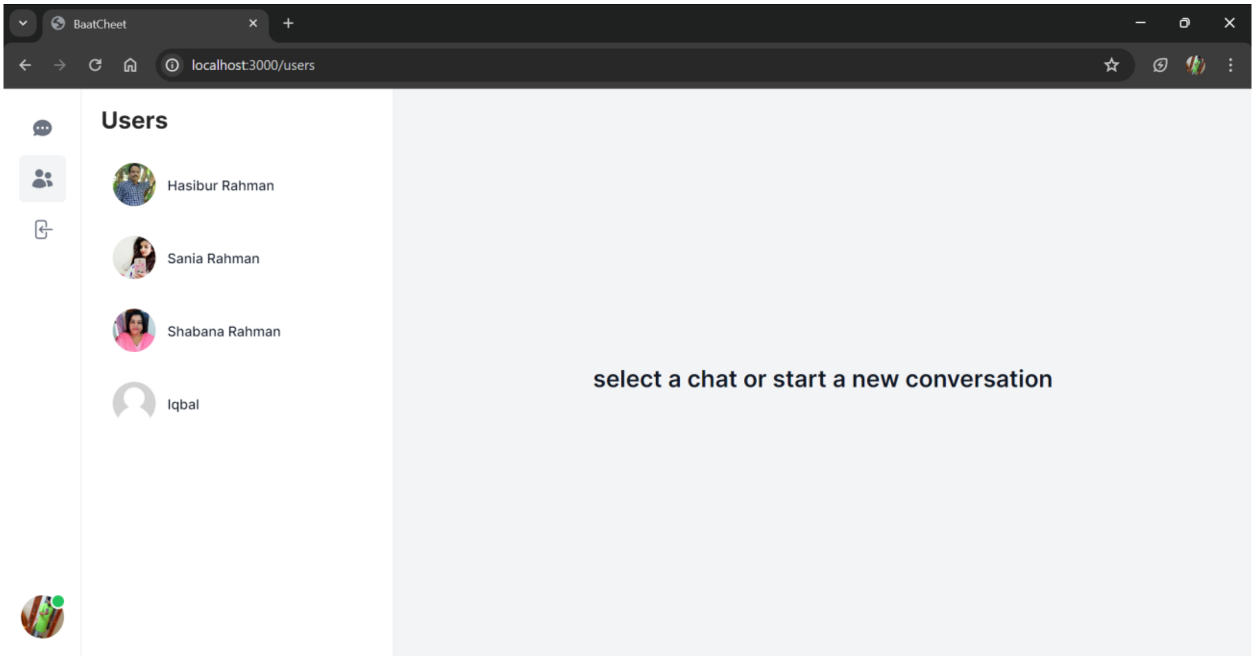
Output:



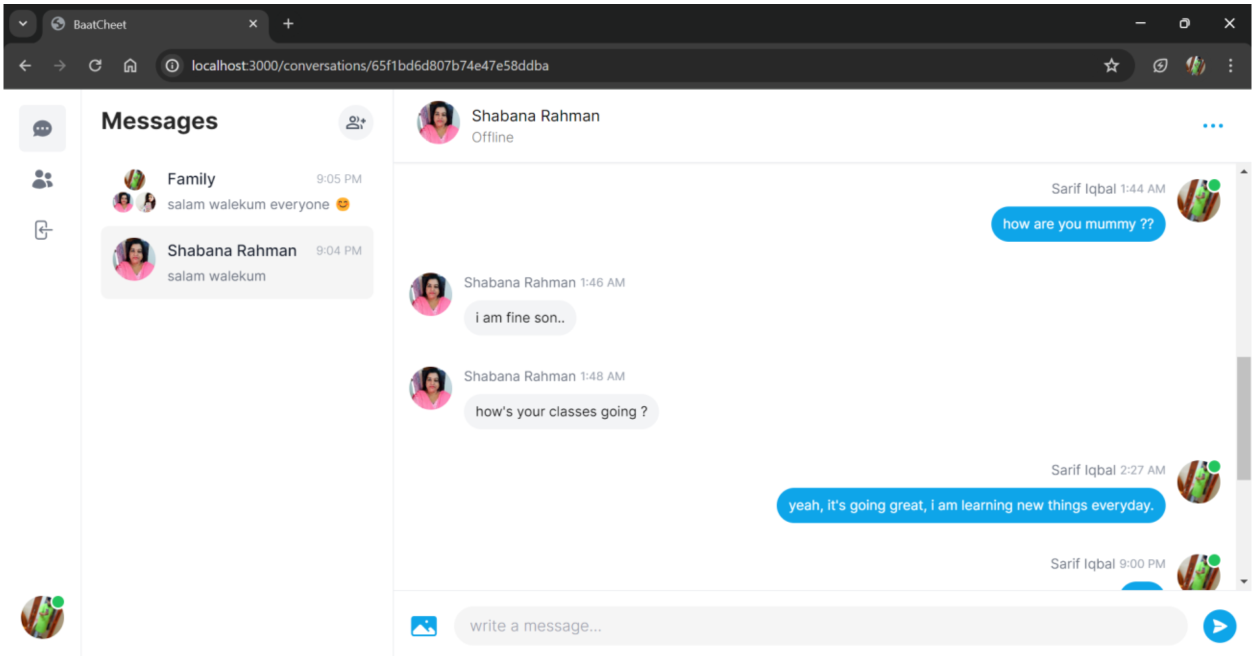
sign in to your account

Don't have an account? Sign up

made with ❤ by Sarif Iqbal



select a chat or start a new conversation



Family 9:05 PM
salam walekum everyone 😊

Shabana Rahman 9:04 PM
salam walekum

Shabana Rahman
Offline

Sarif Iqbal 1:44 AM
how are you mummy ??

Shabana Rahman 1:46 AM
i am fine son..

Shabana Rahman 1:48 AM
how's your classes going ?

Sarif Iqbal 2:27 AM
yeah, it's going great, i am learning new things everyday.

Sarif Iqbal 9:00 PM

Experiment – 14

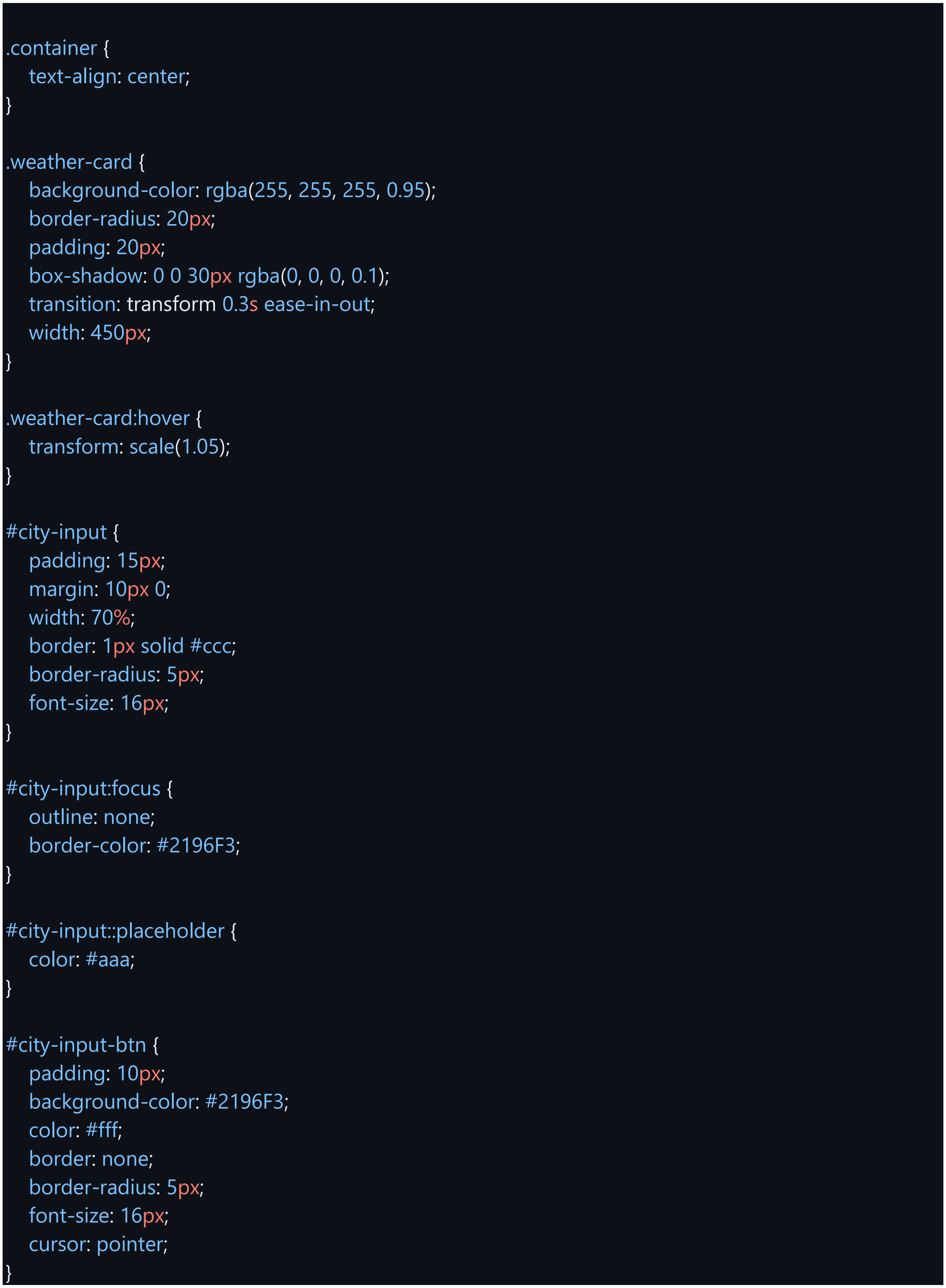
Aim: Weather Forecasting App

Code:

```
<!DOCTYPE html>
<head>
   <link rel="stylesheet" href="style.css">
   <link rel="stylesheet" href=
"https://cdnjs.cloudflare.com/ajax/libs/animate.css/4.1.1/animate.min.css">
   <link rel="stylesheet" href=
"https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.1/css/all.min.css">
   <link rel="stylesheet" href=
"https://fonts.googleapis.com/css2?family=Montserrat:wght@400;700&display=swap">
   <title>GFG App</title>
```

```
</head>

<body>
    <div class="container">
        <div class="weather-card">
            <h1 style="color: turquoise;">
                Sarif Iqbal's Weather App
            </h1>
            <input type="text" id="city-input"
                placeholder="Enter city name">
            <button id="city-input-btn"
                    onclick="weatherFn($('#city-input').val())">
                    Get Weather
            </button>
            <div id="weather-info"
                class="animate__animated animate__fadeIn">
                <h3 id="city-name"></h3>
                <p id="date"></p>
                <p id="temperature"></p>
                <p id="description"></p>
                <p id="wind-speed"></p>
            </div>
        </div>
    </div>
    <script src=
"https://code.jquery.com/jquery-3.6.0.min.js">
    </script>
    <script src=
"https://momentjs.com/downloads/moment.min.js">
    </script>
    <script src="app.js"></script>
</body>

</html>
```

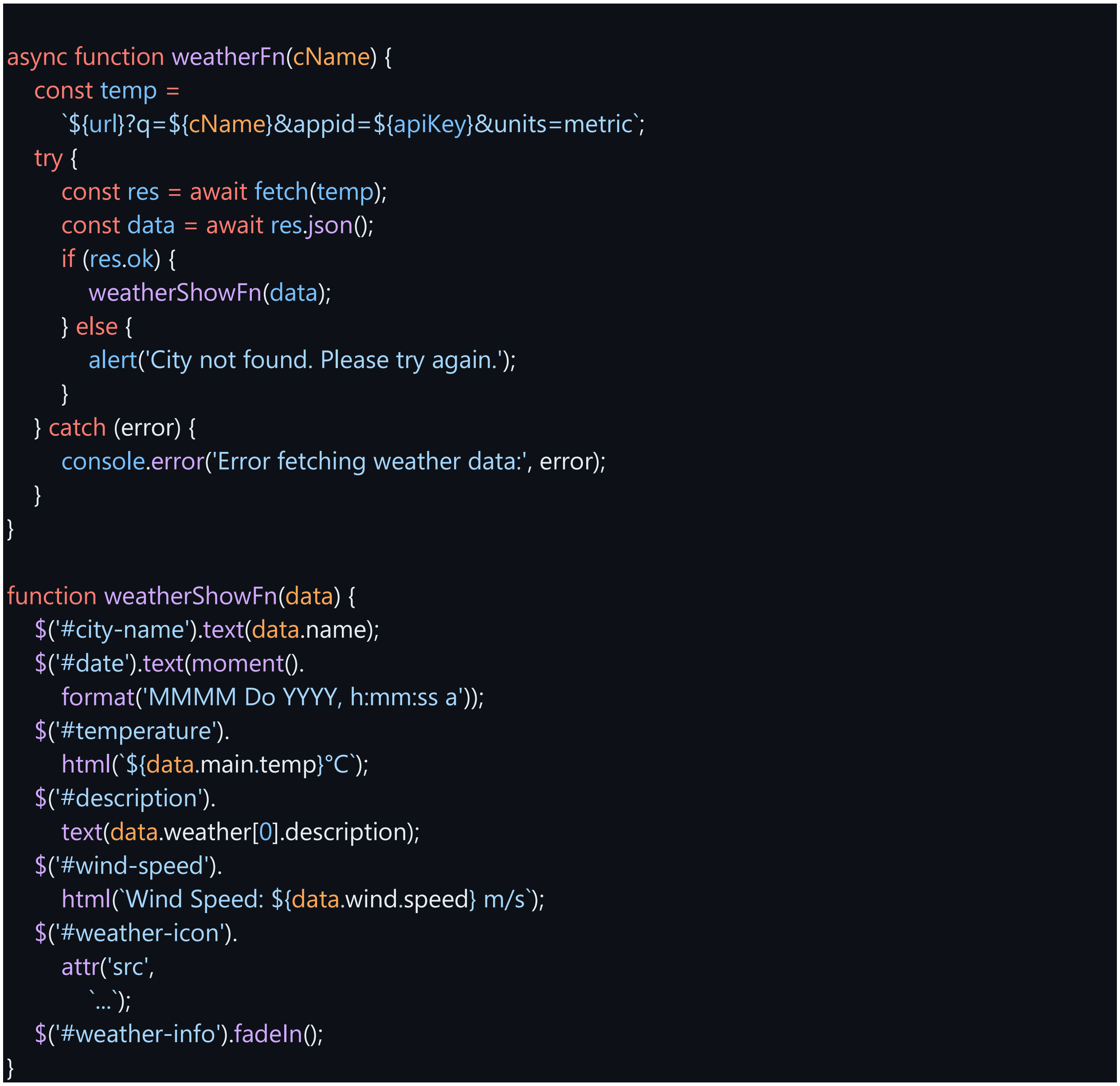
```
body {
    margin: 0;
    font-family: 'Montserrat', sans-serif;
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;
    background: linear-gradient(to right, #1565C0, #2196F3);
}
```



```css
.container {
    text-align: center;
}

.weather-card {
    background-color: rgba(255, 255, 255, 0.95);
    border-radius: 20px;
    padding: 20px;
    box-shadow: 0 0 30px rgba(0, 0, 0, 0.1);
    transition: transform 0.3s ease-in-out;
    width: 450px;
}

.weather-card:hover {
    transform: scale(1.05);
}

#city-input {
    padding: 15px;
    margin: 10px 0;
    width: 70%;
    border: 1px solid #ccc;
    border-radius: 5px;
    font-size: 16px;
}

#city-input:focus {
    outline: none;
    border-color: #2196F3;
}

#city-input::placeholder {
    color: #aaa;
}

#city-input-btn {
    padding: 10px;
    background-color: #2196F3;
    color: #fff;
    border: none;
    border-radius: 5px;
    font-size: 16px;
    cursor: pointer;
}
```

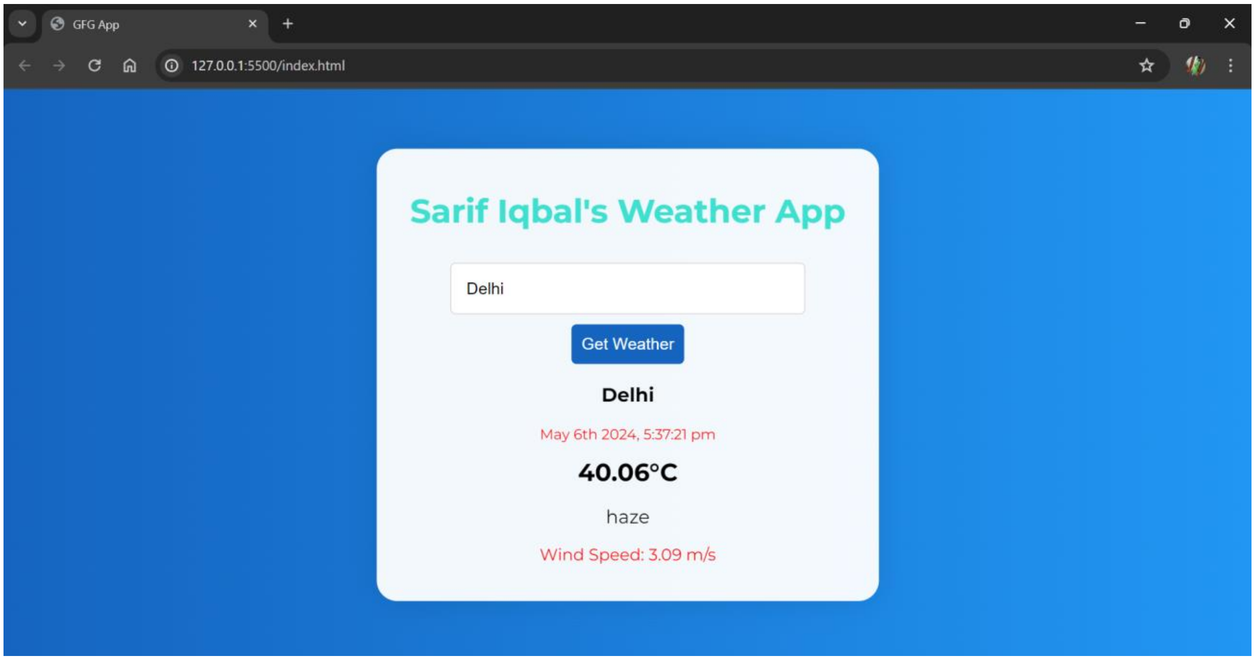
```
#city-input-btn:hover {
    background-color: #1565C0;
}

#weather-info {
    display: none;
}

#weather-icon {
    width: 100px;
    height: 100px;
}

#temperature {
    font-size: 24px;
    font-weight: bold;
    margin: 8px 0;
}

#description {
    font-size: 18px;
    margin-bottom: 10px;
}

#wind-speed {
    font-size: 16px;
    color: rgb(255, 0, 0);
}

#date {
    font-size: 14px;
    color: rgb(255, 0, 0);
}
```

```
const url =
    'https://api.openweathermap.org/data/2.5/weather';
const apiKey =
    'f00c38e0279b7bc85480c3fe775d518c';

$(document).ready(function () {
    weatherFn('Pune');
});
```



```
async function weatherFn(cName) {
    const temp =
        `${url}?q=${cName}&appid=${apiKey}&units=metric`;
    try {
        const res = await fetch(temp);
        const data = await res.json();
        if (res.ok) {
            weatherShowFn(data);
        } else {
            alert('City not found. Please try again.');
        }
    } catch (error) {
        console.error('Error fetching weather data:', error);
    }
}

function weatherShowFn(data) {
    $('#city-name').text(data.name);
    $('#date').text(moment().
        format('MMMM Do YYYY, h:mm:ss a'));
    $('#temperature').
        html(`${data.main.temp}°C`);
    $('#description').
        text(data.weather[0].description);
    $('#wind-speed').
        html(`Wind Speed: ${data.wind.speed} m/s`);
    $('#weather-icon').
        attr('src',
            `...`);
    $('#weather-info').fadeIn();
}
```

Output:



Sarif Iqbal's Weather App

Delhi

Get Weather

Delhi

May 6th 2024, 5:37:21 pm

40.06°C

haze

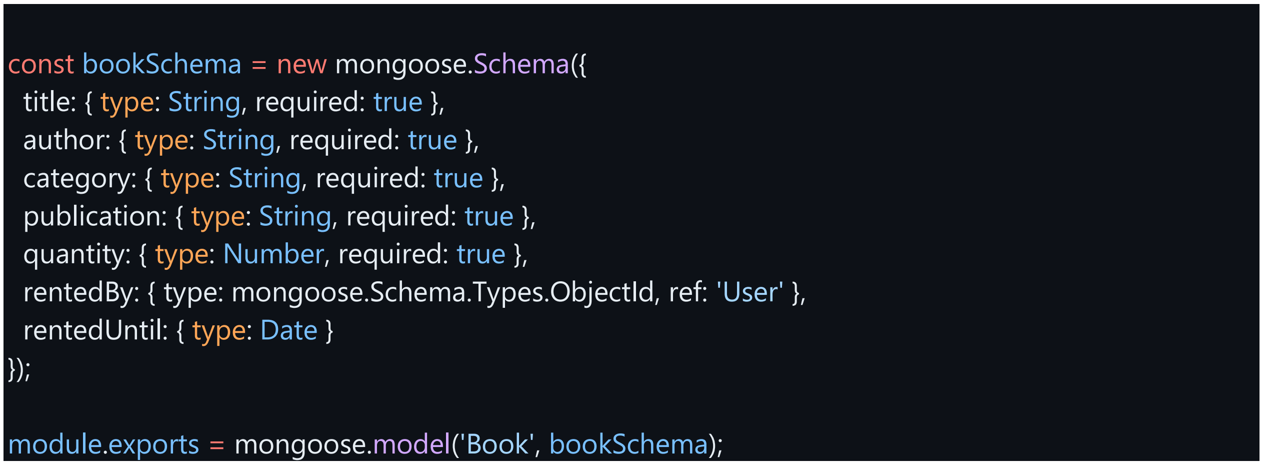
Wind Speed: 3.09 m/s

Experiment – 15

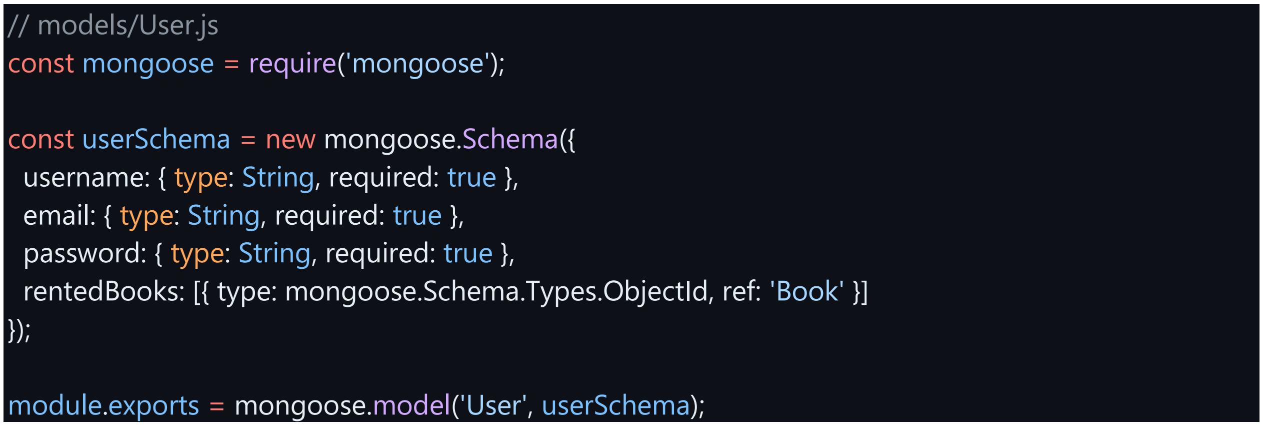
Aim: Bookstore Library and Stock-Keeping App

Code:

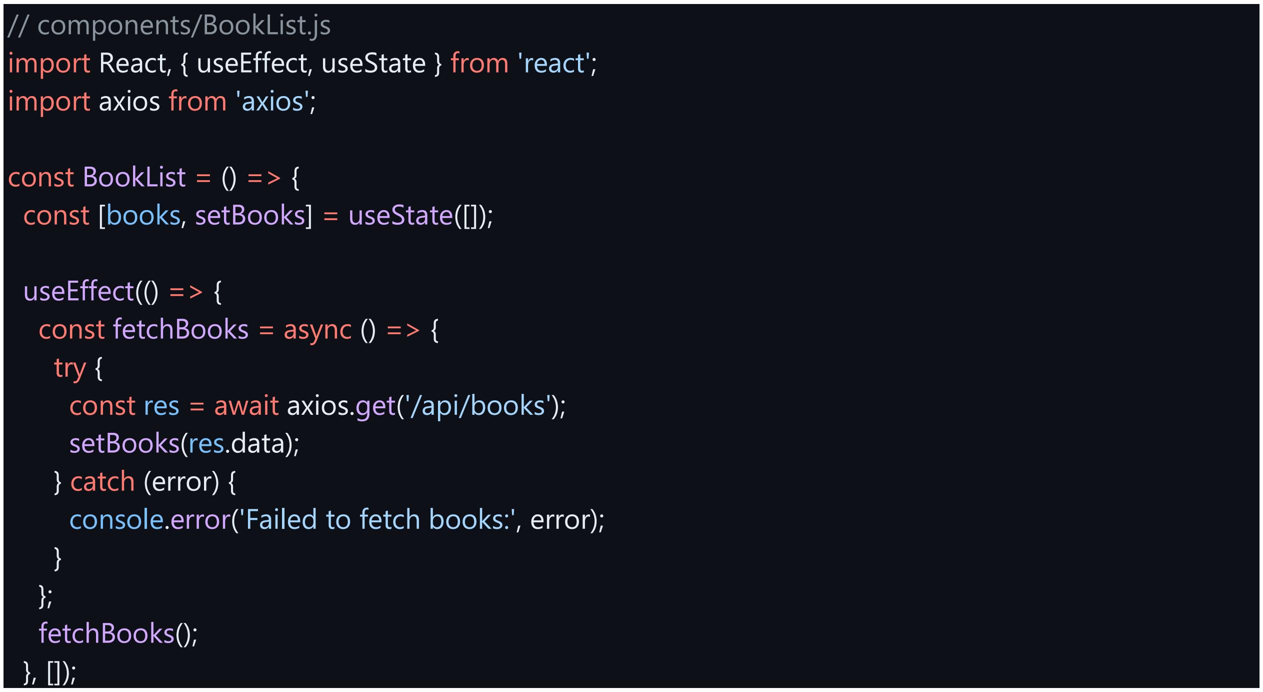
```
// models/Book.js
const mongoose = require('mongoose');
```



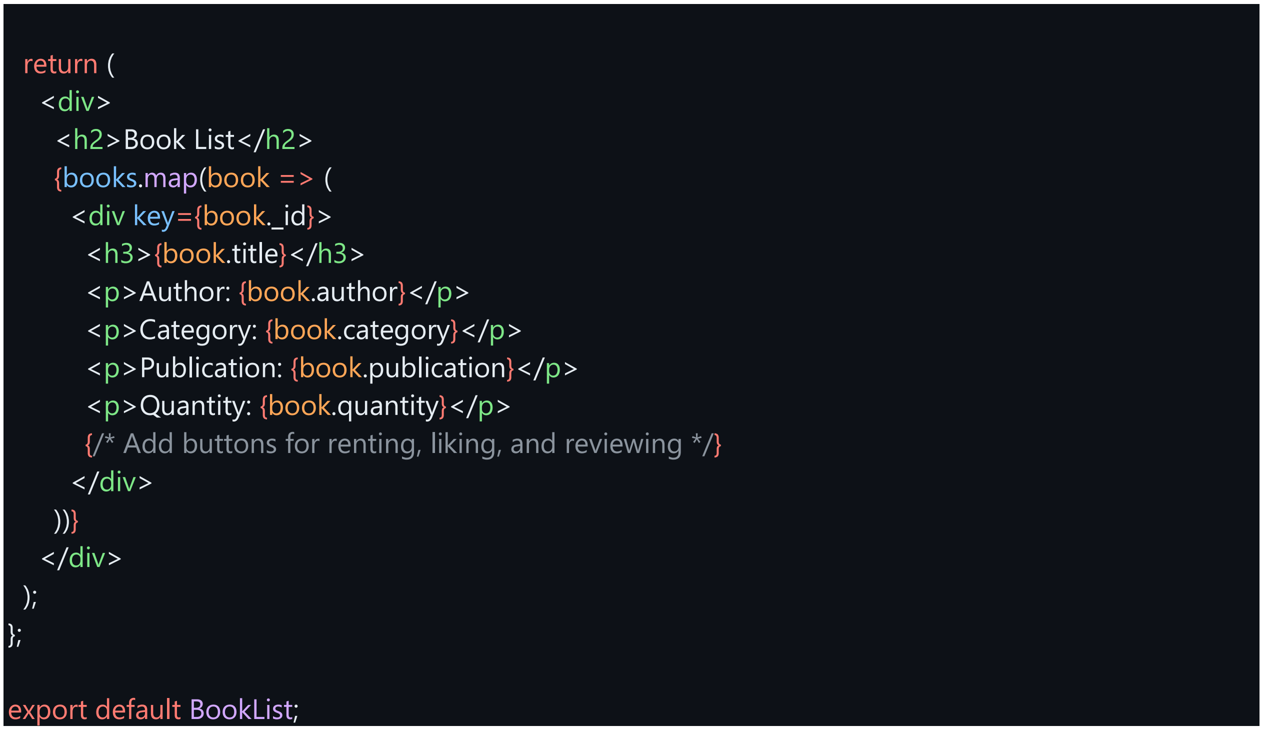
```
const bookSchema = new mongoose.Schema({
  title: { type: String, required: true },
  author: { type: String, required: true },
  category: { type: String, required: true },
  publication: { type: String, required: true },
  quantity: { type: Number, required: true },
  rentedBy: { type: mongoose.Schema.Types.ObjectId, ref: 'User' },
  rentedUntil: { type: Date }
});

module.exports = mongoose.model('Book', bookSchema);
```



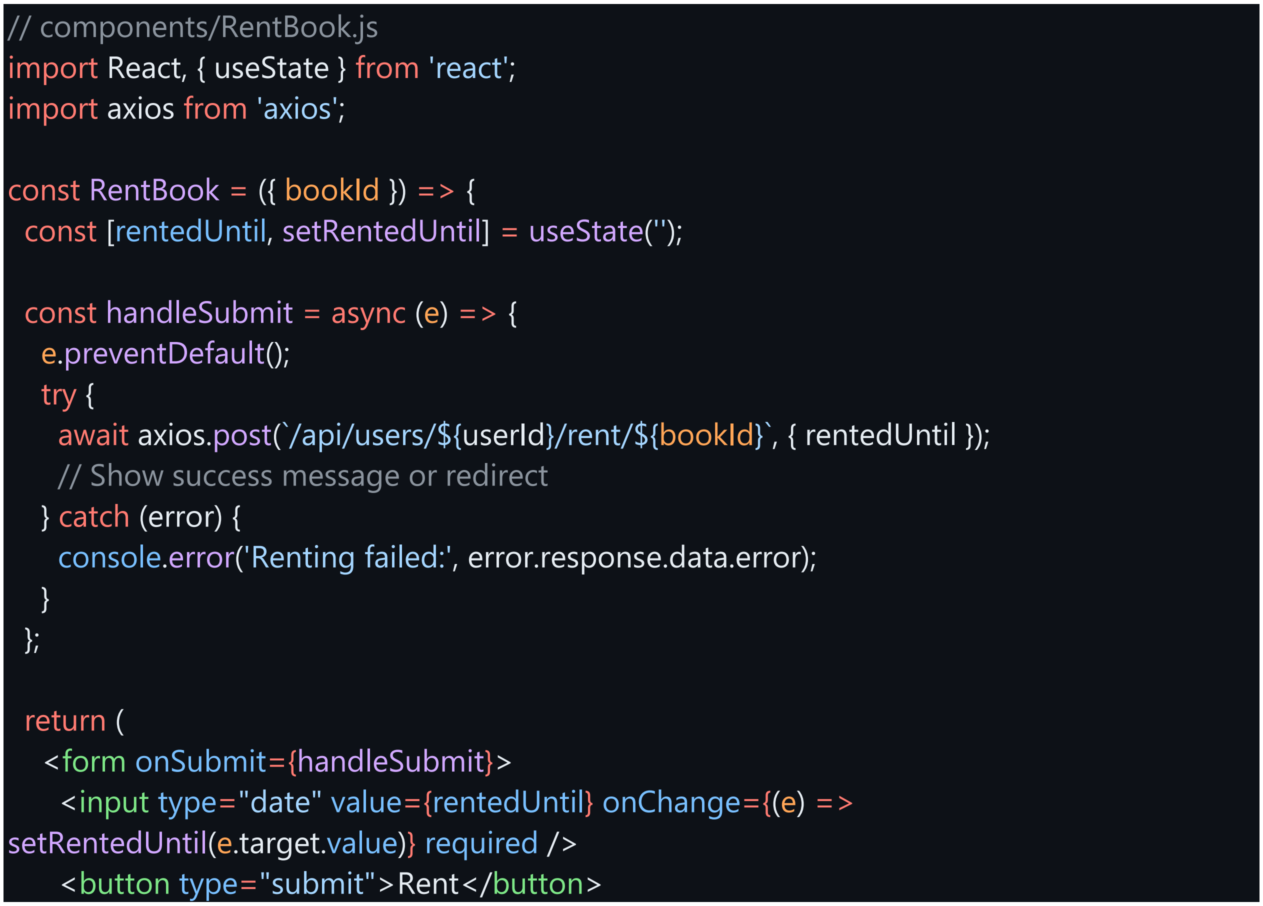
```
// models/User.js
const mongoose = require('mongoose');

const userSchema = new mongoose.Schema({
  username: { type: String, required: true },
  email: { type: String, required: true },
  password: { type: String, required: true },
  rentedBooks: [{ type: mongoose.Schema.Types.ObjectId, ref: 'Book' }]
});

module.exports = mongoose.model('User', userSchema);
```



```
// components/BookList.js
import React, { useEffect, useState } from 'react';
import axios from 'axios';

const BookList = () => {
  const [books, setBooks] = useState([]);

  useEffect(() => {
    const fetchBooks = async () => {
      try {
        const res = await axios.get('/api/books');
        setBooks(res.data);
      } catch (error) {
        console.error('Failed to fetch books:', error);
      }
    };
    fetchBooks();
  }, []);
```



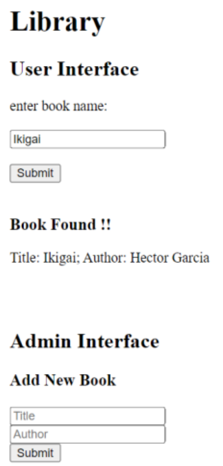
```
  return (
    <div>
      <h2>Book List</h2>
      {books.map(book => (
        <div key={book._id}>
          <h3>{book.title}</h3>
          <p>Author: {book.author}</p>
          <p>Category: {book.category}</p>
          <p>Publication: {book.publication}</p>
          <p>Quantity: {book.quantity}</p>
          {/* Add buttons for renting, liking, and reviewing */}
        </div>
      ))}
    </div>
  );
};

export default BookList;
```



```
// components/RentBook.js
import React, { useState } from 'react';
import axios from 'axios';

const RentBook = ({ bookId }) => {
  const [rentedUntil, setRentedUntil] = useState('');

  const handleSubmit = async (e) => {
    e.preventDefault();
    try {
      await axios.post(`/api/users/${userId}/rent/${bookId}`, { rentedUntil });
      // Show success message or redirect
    } catch (error) {
      console.error('Renting failed:', error.response.data.error);
    }
  };

  return (
    <form onSubmit={handleSubmit}>
      <input type="date" value={rentedUntil} onChange={(e) =>
setRentedUntil(e.target.value)} required />
      <button type="submit">Rent</button>
    </form>
  );
};

export default RentBook;
```

```
// components/Admin/ManageBooks.js
import React, { useState, useEffect } from 'react';
import axios from 'axios';

const ManageBooks = () => {
  const [books, setBooks] = useState([]);

  useEffect(() => {
    const fetchBooks = async () => {
      try {
        const res = await axios.get('/api/books');
        setBooks(res.data);
      } catch (error) {
        console.error('Failed to fetch books:', error);
      }
    };
    fetchBooks();
  }, []);

  const handleDelete = async (bookId) => {
    try {
      await axios.delete(`/api/books/${bookId}`);
      // Show success message or refresh book list
    } catch (error) {
      console.error('Deletion failed:', error.response.data.error);
    }
  };

  return (
    <div>
      <h2>Manage Books</h2>
      {books.map(book => (
        <div key={book._id}>
          <h3>{book.title}</h3>
          <p>Author: {book.author}</p>
          <p>Category: {book.category}</p>
          <p>Publication: {book.publication}</p>
          <p>Quantity: {book.quantity}</p>
          <button onClick={() => handleDelete(book._id)}>Delete</button>
        </div>
      ))}
    </div>
  );
};

export default ManageBooks;
```

Output:



Library

User Interface

enter book name:

Book Found !!

Title: Ikigai; Author: Hector Garcia

Admin Interface

Add New Book