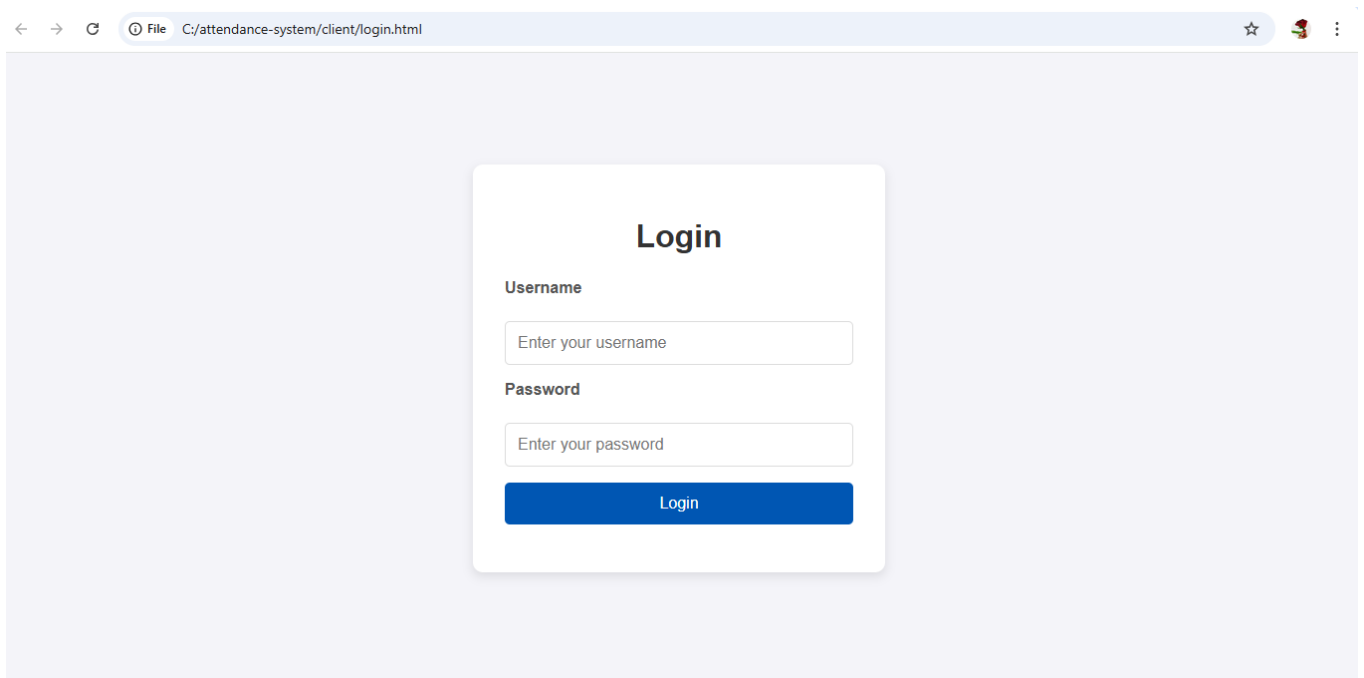


# Practical Assessment for Software Engineer Intern



← → ↻ 📁 File C:/attendance-system/client/login.html ☆ 🌐 ⋮

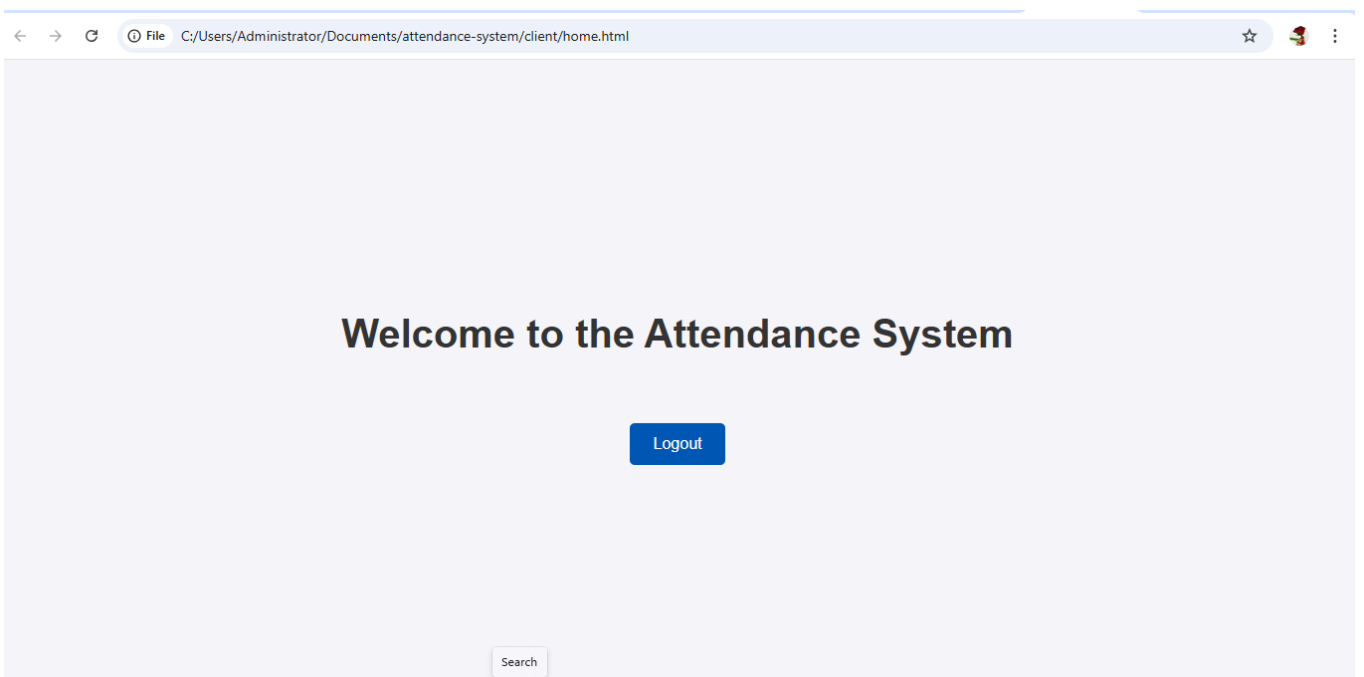
## Login

**Username**

**Password**

Login

#1, created login page



← → ↻ 📁 File C:/Users/Administrator/Documents/attendance-system/client/home.html ☆ 🌐 ⋮

## Welcome to the Attendance System

Logout

Search

#2, created home page

The screenshot shows the Visual Studio Code editor with the 'attendance-system [Administrator]' window. The Explorer sidebar on the left shows the project structure, with the 'client' directory expanded and 'app.js' selected. The main editor displays the content of 'app.js', which is a JavaScript file for a client-side application. The code includes a constant for the API URL, a login function using the Fetch API, and a logout function. The status bar at the bottom indicates the cursor is at line 63, column 4.

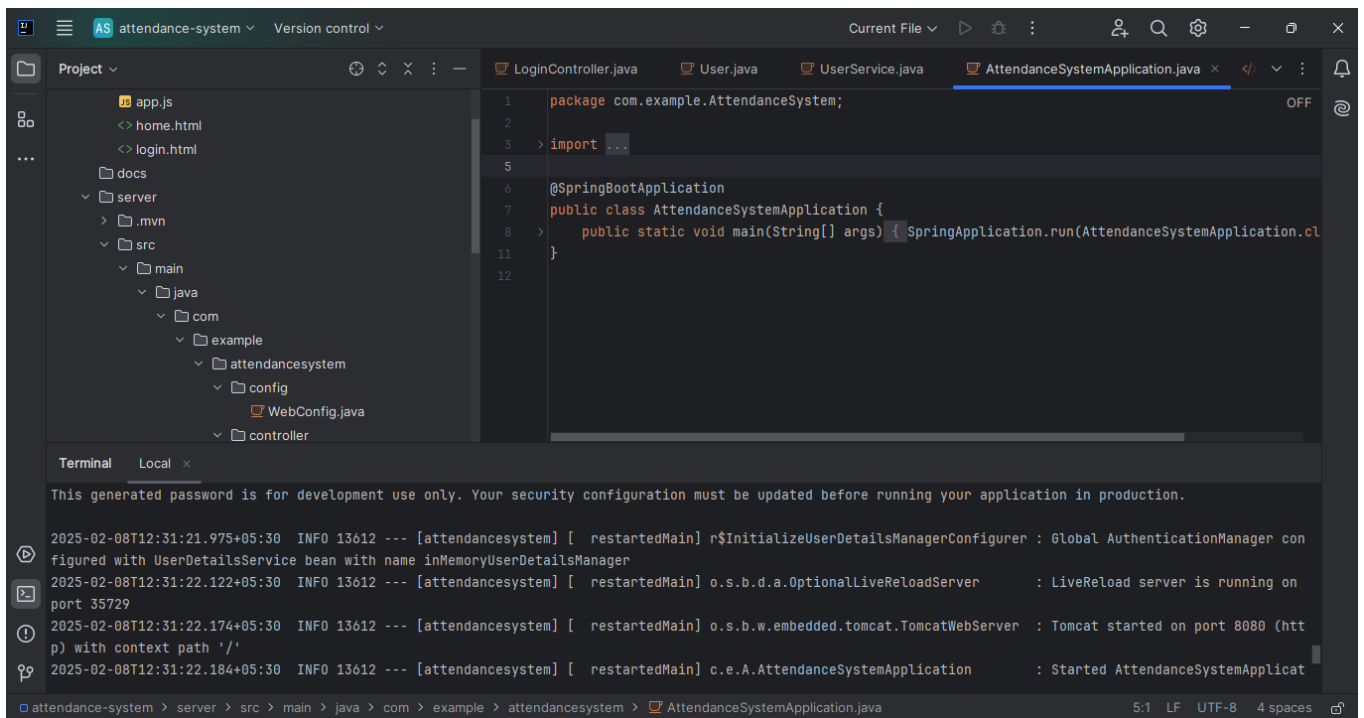
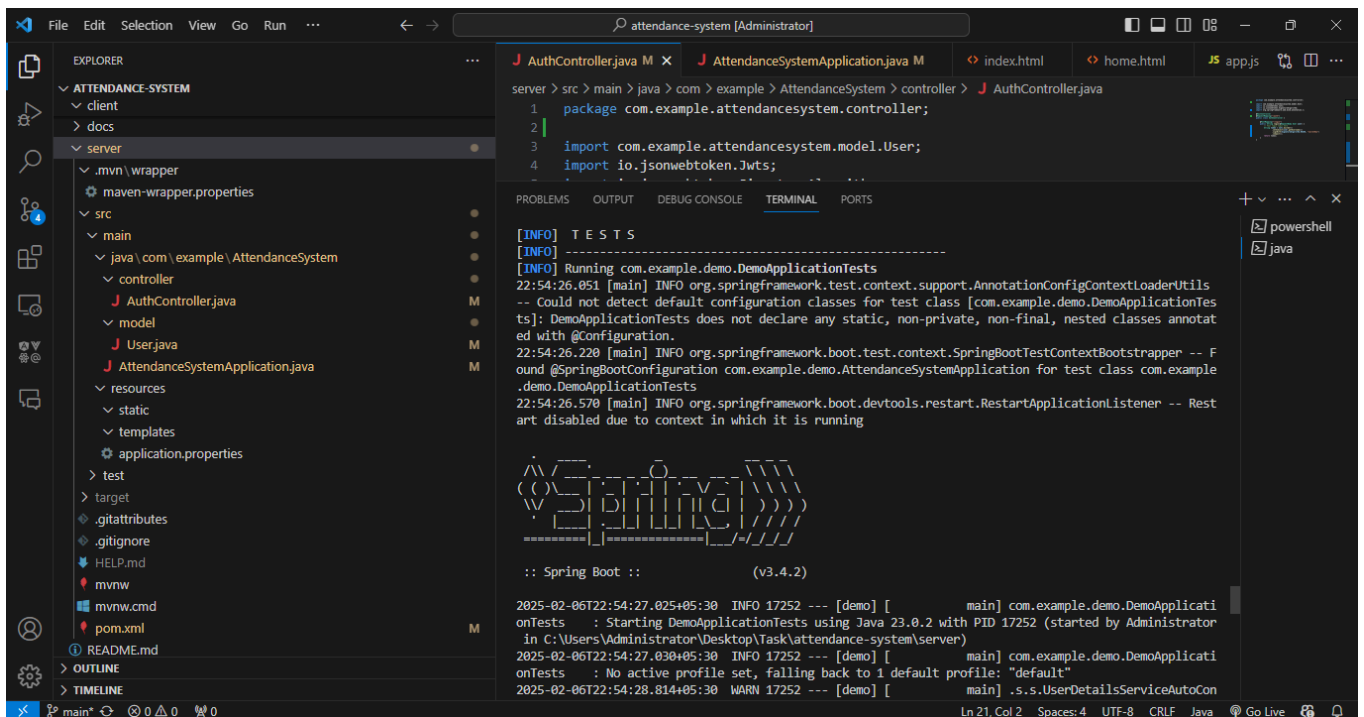
```
client > JS app.js > ...
1  const API_URL = 'http://localhost:8080/api';
2
3  // Login function
4  async function login(username, password) {
5      try {
6          const response = await fetch(`${API_URL}/login`, {
7              method: 'POST',
8              headers: {
9                  'Content-Type': 'application/json',
10             },
11             mode: 'cors',
12             body: JSON.stringify({ username, password }),
13         });
14
15         if (response.ok) {
16             const data = await response.text();
17             localStorage.setItem('token', data); // Store the token
18             window.location.href = 'home.html'; // Redirect to home page
19         } else {
20             throw new Error('Invalid username or password');
21         }
22     } catch (error) {
23         document.getElementById('message').textContent = error.message;
24     }
25 }
26
27 // Logout function
28 function logout() {
29     localStorage.removeItem('token'); // Remove the token
30     window.location.href = 'index.html'; // Redirect to login page
31 }
32
```

#3, created app.js

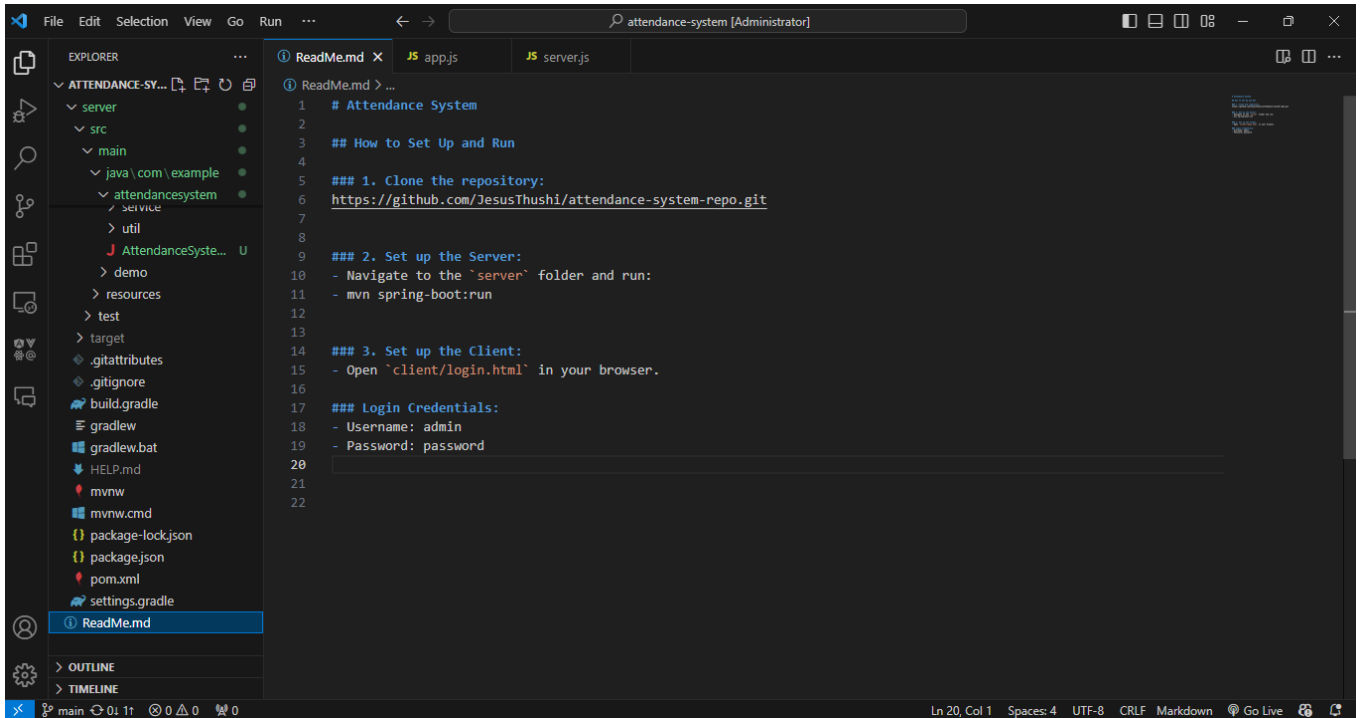
The screenshot shows the Visual Studio Code editor with the 'attendance-system [Administrator]' window. The Explorer sidebar on the left shows the project structure, with the 'server' directory expanded and 'server.js' selected. The main editor displays the content of 'server.js', which is a JavaScript file for a server-side application. The code includes a simple API route for login, static file serving, and a server listening on port 5000. The status bar at the bottom indicates the cursor is at line 20, column 1.

```
client > JS server.js > ...
9      origin: 'http://localhost:5500/login.html',
10     methods: ['GET', 'POST', 'PUT', 'DELETE'],
11     allowedHeaders: ['Content-Type', 'Authorization'],
12     credentials: true,
13 });
14
15
16 app.listen(5000, () => {
17     console.log('Server running on port 5000');
18 });
19
20
21 // Serve static files
22 app.use(express.static(path.join(__dirname, 'client')));
23
24 // Simple API route for demonstration
25 app.use(express.json());
26 app.post('/api/login', (req, res) => {
27     const { username, password } = req.body;
28     if (username === 'admin' && password === 'password') {
29         res.json({ token: 'mock-jwt-token' });
30     } else {
31         res.status(401).json({ message: 'Invalid credentials' });
32     }
33 });
34
35 const PORT = 8080;
36 app.listen(PORT, () => {
37     console.log(`Server running at http://localhost:${PORT}`);
38 });
39
```

#4, created server.js



#5, server code initialized with Spring Boot and implemented login REST API with JWT token



#6, added ReadMe.md file with project setup instructions

Name :-J.Thusyanthiny

Mobile No :- +94 740087469

Email Id :- [thushijeyaseelan25@gmail.com](mailto:thushijeyaseelan25@gmail.com)

Repository link :- <https://github.com/JesusThushi/attendance-system-repo.git>