॥वसुधैव कुटुम्बकम्॥

FULL STACK DEVLOPMENT

ASSIGNMENT - 9

Prepared by:

Apoorv Gupta

21070126018 (AIML A1)

In this assignment, learn about mongodb and node js in order to combine functionality and record data from web to store in a nosql data storage system.

Description

The Login System project is a web application developed using Node.js, Express, and MongoDB, enabling users to securely log in with their credentials or create new accounts if they don't have one. The application ensures secure user authentication by storing passwords encrypted in the MongoDB database and provides intuitive feedback messages based on login outcomes. It incorporates proper error handling and validation mechanisms to maintain data integrity and prevent security vulnerabilities. The web interface, designed with HTML, CSS, and Handlebars (HBS) templates, offers user-friendly forms for login and signup, enhancing the overall user experience.

In terms of architecture, the project follows a RESTful API design, with Express routes handling login and signup functionality, interacting with the MongoDB database through Mongoose models. On the client side, JavaScript is utilised for front-end logic, including form submissions and message display. The project fosters modularity and maintainability by organising code into separate JavaScript files for login and signup pages, enhancing readability and ease of maintenance. Overall, the Login System project exemplifies a robust, scalable, and user-friendly approach to user authentication in web applications, leveraging modern technologies and best practices in web development.

App.js

```
// app.js

const express = require('express');
const mongoose = require('mongoose');
const bodyParser = require('body-parser');
const path = require('path');
const authRoutes = require('./routes/auth');

const app = express();

// Middleware
app.use(bodyParser.urlencoded({ extended: false }));
app.use(bodyParser.json());
app.use(express.static(path.join(__dirname, 'views')));

// Connect to MongoDB
mongoose.connect('mongodb://localhost:27017/loginSystem', {
  useNewUrlParser: true,
  useUnifiedTopology: true
})
.then(() => console.log('MongoDB connected'))
.catch(err => console.error(err));

// Routes
app.use('/auth', authRoutes);

// Route for serving the login page
app.get('/', (req, res) => {
  res.sendFile(path.join(__dirname, 'views', 'login.html'));
});

// Route for serving the signup page
app.get('/signup', (req, res) => {
  res.sendFile(path.join(__dirname, 'views', 'signup.html'));
});

// Redirect any other requests to the root URL to the login page
app.get('*', (req, res) => {
  res.redirect('/');
});

// Start server
const PORT = process.env.PORT || 3000;
app.listen(PORT, () => console.log(`Server running on port ${PORT}`));
```

auth.js

```
// routes/auth.js

const express = require('express');
const router = express.Router();
const User = require('../models/User');

// Login route
router.post('/login', async (req, res) => {
    const { username, password } = req.body;
    try {
        const user = await User.findOne({ username, password });
        if (user) {
            res.json({ success: true, message: `Hello, ${username}! Login
            successful` });
        } else {
            res.status(401).json({ success: false, message: 'Incorrect
            username or password. Please try again.' });
        }
    } catch (error) {
        res.status(500).json({ success: false, message: 'Internal server
        error' });
    }
});

// Signup route
router.post('/signup', async (req, res) => {
    const { username, password } = req.body;
    try {
        const existingUser = await User.findOne({ username });
        if (existingUser) {
            return res.status(400).json({ success: false, message: 'Username
            already exists. Please choose a different username.' });
        }
        const newUser = new User({ username, password });
        await newUser.save();
        res.status(201).json({ success: true, message: 'Signup successful' });
    } catch (error) {
        res.status(500).json({ success: false, message: 'Internal server
        error' });
    }
});

module.exports = router;
```

User.js

```
// models/user.js

const mongoose = require('mongoose');

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

const User = mongoose.model('User', userSchema);

module.exports = User;
```

MongoDB

```
loginSystem> db.users.find()
[
  {
    _id: ObjectId('661ad6a57d7dfc518316c9b5'),
    username: 'apoorv',
    password: 'apoorv'
  },
  {
    _id: ObjectId('661b9caa653eeb60f6ec4c21'),
    username: 'apoorv2',
    password: 'apoorv2',
    __v: 0
  }
]
```

login.html

```html
<!-- views/login.html -->

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Login</title>
    <link rel="stylesheet" href="styles.css">
</head>
<body>
    <div class="login-container">
        <form id="login-form">
            <h2>Login</h2>
            <div class="form-group">
                <label for="username">Username:</label>
                <input type="text" id="username" name="username" required>
            </div>
            <div class="form-group">
                <label for="password">Password:</label>
                <input type="password" id="password" name="password" required>
            </div>
            <button type="submit">Login</button>
            <p id="error-message" class="error-message"></p>
            <p>Don't have an account? <a href="/signup">Sign Up</a></p>
        </form>
    </div>

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

signup.html

```html
<!-- views/signup.html -->

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Sign Up</title>
    <link rel="stylesheet" href="styles.css">
</head>
<body>
    <div class="signup-container">
        <form id="signup-form">
            <h2>Sign Up</h2>
            <div class="form-group">
                <label for="username">Username:</label>
                <input type="text" id="username" name="username" required>
            </div>
            <div class="form-group">
                <label for="password">Password:</label>
                <input type="password" id="password" name="password" required>
            </div>
            <button type="submit">Sign Up</button>
            <p id="error-message" class="error-message"></p>
            <p>Already have an account? <a href="/login">Login</a></p>
        </form>
    </div>

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

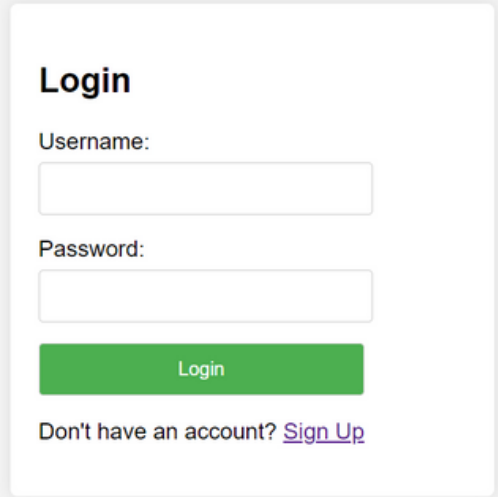
login.js

```
// login.js

document.addEventListener('DOMContentLoaded', function() {
    const loginForm = document.getElementById('login-form');
    const errorMessage = document.getElementById('error-message');

    loginForm.addEventListener('submit', async function(event) {
        event.preventDefault();
        const username = document.getElementById('username').value;
        const password = document.getElementById('password').value;

        try {
            const response = await fetch('/auth/login', {
                method: 'POST',
                headers: { 'Content-Type': 'application/json' },
                body: JSON.stringify({ username, password })
            });
            const data = await response.json();
            if (!response.ok) {
                throw new Error(data.message);
            }
            // Display success message
            errorMessage.textContent = data.message;
        } catch (error) {
            errorMessage.textContent = error.message;
        }
    });
```

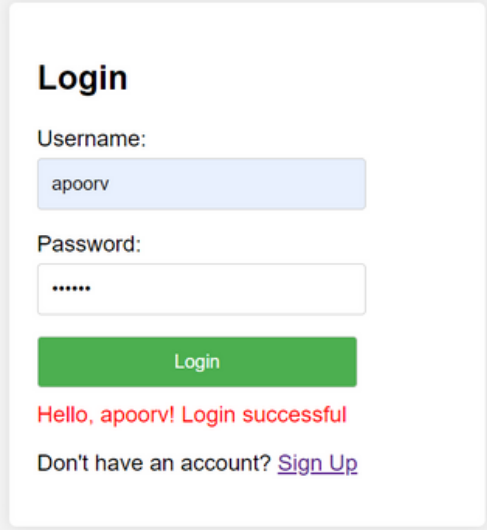
signup.js

```
// signup.js

document.addEventListener('DOMContentLoaded', function() {
    const signupForm = document.getElementById('signup-form');
    const errorMessage = document.getElementById('error-message');

    signupForm.addEventListener('submit', async function(event) {
        event.preventDefault();
        const username = document.getElementById('username').value;
        const password = document.getElementById('password').value;

        try {
            const response = await fetch('/auth/signup', {
                method: 'POST',
                headers: { 'Content-Type': 'application/json' },
                body: JSON.stringify({ username, password })
            });
            const data = await response.json();
            if (!response.ok) {
                throw new Error(data.message);
            }
            // Display success message
            errorMessage.textContent = data.message;
        } catch (error) {
            errorMessage.textContent = error.message;
        }
    });
});
```

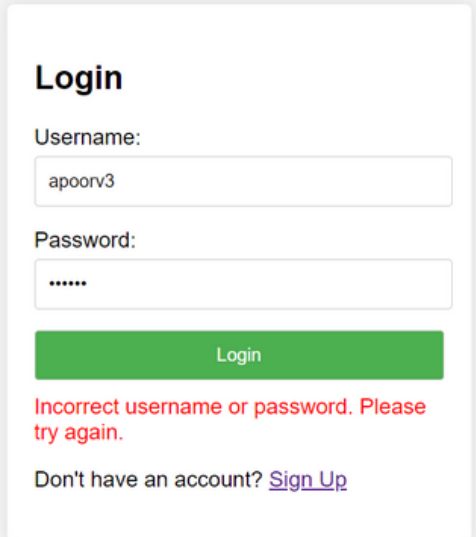
Output login



Landing login page

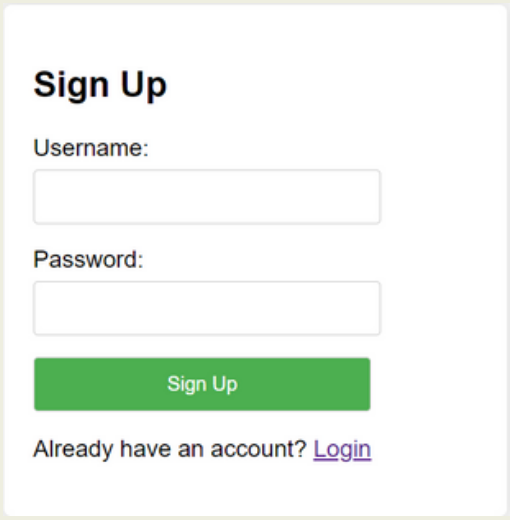


Message with username for successful login

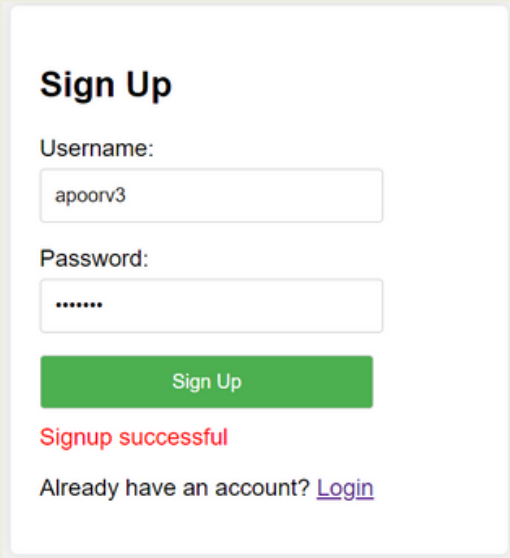


Error message for unsuccessful login

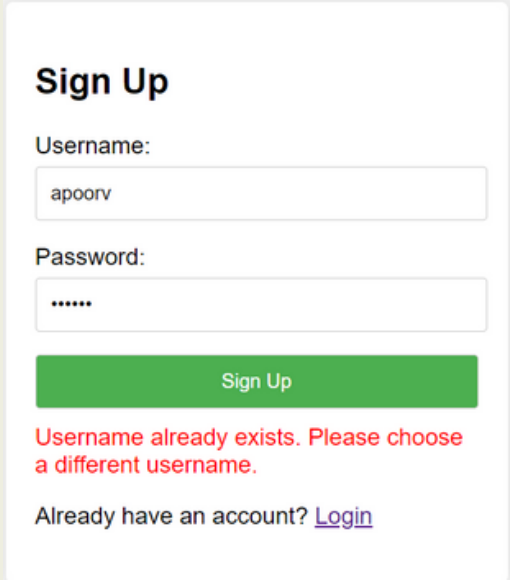
Output signup



Landing signup page



Message with username for successful Registration



Error message for existing credentials in database

About Me

Hi, I am Apoorv
I study at symbiosis pune, branch as AIML.

PRN: 21070126018
Batch: 2021-2025
Division: AIML-A1