

**Shri Ramdeobaba College of Engineering & Management Nagpur-13**

**Department of Computer Application**

**Session: 2023-2024**



**Submission for**

**Course Name:** Full Stack Web Development using MEAN

**Course Code:** MCP548

**Name of the Student:** Jayesh Lalit Nandanwar

**Class Roll No:** 26

**Semester:** MCA II semester

**Shift:** 2

**Batch:** 2

## Assignment 2

**Aim:** Create the website for Cinemax by showing the current running movies and the next week's scheduled movies. Admin can enter all the information like movie title, show timings, cast and other details so that it is neatly displayed on website.

**Code:**

- **app.js:**

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

// Middlewares
app.use(bodyParser.urlencoded({ extended: true }));
app.use(express.static('public'));
app.set('view engine', 'ejs');
app.use(session({
  secret: 'secret-key',
  resave: false,
  saveUninitialized: true
}));

// Routes
const movieController = require('./controllers/movieController');
const adminController = require('./controllers/adminController');
const authController = require('./controllers/authController');
app.use('/', movieController);
app.use('/admin', adminController);
app.use('/auth', authController);

// Start server
const PORT = 3001;
app.listen(PORT, () => console.log(`Server started on port ${PORT}`));
```

## Controllers:

- adminController.js

```
const express = require('express');
const router = express.Router();
const Movie = require('../models/movie.js');
const multer = require('multer');
var storage = multer.diskStorage({
  destination: function (req, file, callback) {
    callback(null, './public/uploads");
  },
  filename: function (req, file, callback) {
    callback(null, file.originalname);
  },
});

// Multer configuration for file upload
const upload = multer({ storage: storage }); // Specify the destination directory for file uploads

// Admin page
router.get('/', (req, res) => {
  // Check if user is authenticated
  if (!req.session.authenticated) {
    return res.redirect('/auth/login');
  }
  res.render('admin', { req: req });
});

// Add movie form
router.get('/add', (req, res) => {
  // Check if user is authenticated
  if (!req.session.authenticated) {
    return res.redirect('/auth/login');
  }
  res.render('add_movie', { req: req });
});

// Add movie
router.post('/add', upload.single('poster'), (req, res) => {
  // Check if user is authenticated
  if (!req.session.authenticated) {
    return res.redirect('/auth/login');
  }
  // ... (rest of the code for adding a movie)
});
```

```

    }

    // Extract form data
    const { title, storyline, timing, cast, day } = req.body;
    // File path of the uploaded poster
    const posterFilePath = req.file.path.split("\\")[1] + "\\" + req.file.path.split("\\")[2];
    console.log(posterFilePath);

    // Add movie to database
    Movie.addMovie(title, storyline, timing, cast, day, posterFilePath)
    console.log('Movie added successfully. ');
    res.json(200, "Movie added successfully.")
  });

  module.exports = router;

```

#### • authController.js

```

const express = require('express');
const router = express.Router();
const fs = require('fs');

// Load users from JSON file
const usersData = JSON.parse(fs.readFileSync('./data/users.json', 'utf-8'));

// Login page
router.get('/login', (req, res) => {
  res.render('login', { req: req });
});

// Authenticate user
router.post('/login', (req, res) => {
  const { username, password } = req.body;
  // Check if user exists and password matches
  const user = usersData.users.find(user => user.username === username &&
  user.password === password);
  if (user) {
    req.session.authenticated = true; // Set session flag for authentication
    res.redirect('/admin');
  } else {
    res.redirect('/auth/login');
  }
});

```

```

    }
  });

  // Login page
  router.get('/signup', (req, res) => {
    res.render('registration', { req: req });
  });

  // Authenticate user
  router.post('/signup', (req, res) => {
    const { fullName, username, password } = req.body;
    // Check if username already exists
    const userExists = usersData.users.find(user => user.username === username);
    if (userExists) {
      res.redirect('/auth/signup');
    } else {
      // Add new user to usersData
      usersData.users.push({ fullName, username, password });
      // Write updated usersData to users.json file
      // console.log(JSON.stringify(usersData, null, 2))
      fs.writeFileSync('./data/users.json', JSON.stringify(usersData, null, 2));
      res.redirect('/auth/login');
    }
  });

  // Logout
  router.get('/logout', (req, res) => {
    req.session.destroy(); // Destroy session
    res.redirect('/');
  });

  module.exports = router;

```

#### • movieController.js

```

const express = require('express');
const router = express.Router();
const Movie = require('../models/movie.js');

// Home page
router.get('/', (req, res) => {

```

```

    // Fetch current and upcoming movies from database
    const currentMovies = Movie.getCurrentMovies();
    const upcomingMovies = Movie.getUpcomingMovies();
    res.render('index', { currentMovies, upcomingMovies ,req: req});
  });

  // Schedule page
  router.get('/schedule', (req, res) => {
    // Fetch schedule of movies from database
    const schedule = Movie.getSchedule();
    res.render('schedule', { schedule , req: req });
  });

  // Add movie to schedule
  router.post('/schedule/add', (req, res) => {
    const { title, timing, day } = req.body;
    if (title && timing && day) {
      Movie.updateSchedule(day, title, timing);
      res.redirect('/schedule');
    } else {
      res.status(400).send('Bad Request: Missing title, timing, or day.');
```

```

    }
  });

  // Update schedule
  router.post('/schedule/update', (req, res) => {
    const { title, timing, day } = req.body;
    if (title && timing && day) {
      Movie.updateSchedule(day, title, timing);
      res.redirect('/schedule');
    } else {
      res.status(400).send('Bad Request: Missing title, timing, or day.');
```

```

    module.exports = router;
  });
}
```

**data:**

• movies.json

```
{
  "currentMovies": [
    {
      "title": "The Matrix Resurrections",
      "cast": "Keanu Reeves, Carrie-Anne Moss",
      "poster":
        "https://www.themoviedb.org/t/p/w600_and_h900_bestv2/mRVs2fS3cf30W8OsLX7F
        AIvi0QH.jpg",
      "storyline": "Neo returns to the Matrix, still unsure of what is real and what is not, as
        he seeks to uncover the truth about his existence."
    },
    {
      "title": "Spider-Man: No Way Home",
      "cast": "Tom Holland, Zendaya",
      "poster":
        "https://www.themoviedb.org/t/p/w600_and_h900_bestv2/5weKu49pzJCt06OPpjuT80e
        fnQj.jpg",
      "storyline": "Peter Parker's identity as Spider-Man is revealed, leading to a
        dangerous confrontation with villains from alternate realities, forcing him to seek help
        from Doctor Strange."
    }
  ],
  "upcomingMovies": [
    {
      "title": "Avatar 2",
      "cast": "Sam Worthington, Zoe Saldana",
      "poster":
        "https://www.themoviedb.org/t/p/w600_and_h900_bestv2/t6HIqrRAclMCA60NsSmeq
        e9RmNV.jpg",
      "storyline": "Jake Sully and Neytiri navigate Pandora's lush ecosystem, facing new
        threats while discovering more about the planet's indigenous population."
    },
    {
      "title": "Jurassic World: Dominion",
      "cast": "Chris Pratt, Bryce Dallas Howard",
      "poster":
        "https://www.themoviedb.org/t/p/w600_and_h900_bestv2/kAVRgw7GgK1CfYIEJq8M
        E6EvRIgU.jpg",

```

"storyline": "Dinosaurs roam freely on mainland USA, threatening human existence, as former Jurassic World inhabitants must find a way to coexist with these ancient creatures."

},

{

"title": "Wonder Woman 1984",

"cast": "Gal Gadot, Chris Pine",

"poster":

"https://www.themoviedb.org/t/p/w600\_and\_h900\_bestv2/8UIWHLMpgZm9bx6QYh0NFoq67TZ.jpg",

"storyline": "Diana Prince faces off against Maxwell Lord and the Cheetah, while dealing with the return of her long-lost love, Steve Trevor, in the vibrant 1980s."

},

{

"title": "Dune",

"cast": "Timothée Chalamet, Zendaya",

"poster":

"https://www.themoviedb.org/t/p/w600\_and\_h900\_bestv2/d5NXSkIXo0qyIYkgV94XAgMickC.jpg",

"storyline": "Paul Atreides, a young nobleman, leads a rebellion to reclaim his homeland from the oppressive rule of House Harkonnen, while also unlocking his own latent powers."

},

{

"title": "Black Widow",

"cast": "Scarlett Johansson, Florence Pugh",

"poster":

"https://www.themoviedb.org/t/p/w600\_and\_h900\_bestv2/kwB7d51AIcyzPOBOHLCEZJkmPhQ.jpg",

"storyline": "Natasha Romanoff confronts her past as a spy and the broken relationships she left behind, while facing a deadly conspiracy linked to her former life."

},

{

"title": "No Time to Die",

"cast": "Daniel Craig, Léa Seydoux",

"poster":

"https://www.themoviedb.org/t/p/w600\_and\_h900\_bestv2/eqWaeh21e4ZgHjwpULZVHCGIq9X.jpg",

"storyline": "James Bond comes out of retirement to face Safin, a new villain armed with dangerous technology, as he races against time to save the world."

},

{

"title": "Oppenheimer",



```

        "cast": "Cillian Murphy, Emily Blunt",
        "poster": "uploads\\Oppenheimer.jpg",
        "storyline": "The story of J. Robert Oppenheimer's role in the development of the
atomic bomb during World War II."
    }
]
}

```

• schedule.json

```

{
  "Monday": [
    {
      "title": "The Matrix Resurrections",
      "timing": "8:00 PM"
    },
    {
      "title": "Spider-Man: No Way Home",
      "timing": "10:00 PM"
    }
  ],
  "Tuesday": [
    {
      "title": "Spider-Man: No Way Home",
      "timing": "10:00 PM"
    },
    {
      "title": "Avatar 2",
      "timing": "8:00 PM"
    },
    {
      "title": "Jurassic World: Dominion",
      "timing": "10:00 PM"
    }
  ],
  "Wednesday": [],
  "Thursday": [],
  "Friday": [
    {
      "title": "Oppenheimer",
      "timing": "15:30"
    }
  ]
}

```

```
],  
  "Saturday": [],  
  "Sunday": []  
}
```

• users.json

```
{  
  "users": [  
    {  
      "username": "admin",  
      "password": "admin123"  
    },  
    {  
      "username": "john_doe",  
      "password": "password123"  
    },  
    {  
      "username": "jane_smith",  
      "password": "cinemax456"  
    }  
  ]  
}
```

## Models:

- Movie.js

```
const moviesData = require('../data/movies.json');
const moviesSchedule = require('../data/schedule.json');

const fs = require('fs');

class Movie {
  static getCurrentMovies() {
    return moviesData.currentMovies;
  }

  static getUpcomingMovies() {
    return moviesData.upcomingMovies;
  }

  static getSchedule() {
    try {
      const scheduleData = fs.readFileSync('../data/schedule.json', 'utf-8');
      return JSON.parse(scheduleData);
    } catch (error) {
      console.error('Error reading schedule:', error);
      return {};
    }
  }

  static updateSchedule(day, movieTitle, timing) {
    try {
      let schedule = this.getSchedule();
      if (schedule[day]) {
        schedule[day].push({ title: movieTitle, timing });
      } else {
        schedule[day] = [{ title: movieTitle, timing }];
      }
      fs.writeFileSync('../data/schedule.json', JSON.stringify(schedule, null, 2));
      console.log('Schedule updated successfully.');
    } catch (error) {
      console.error('Error updating schedule:', error);
    }
  }

  static addMovie(title, storyline, timing, cast, day, posterFilePath) {
```

```
    var poster=posterFilePath;
    moviesData.upcomingMovies.push({ title, cast, poster ,storyline});
    fs.writeFileSync('./data/movies.json', JSON.stringify(moviesData, null, 2));
    moviesSchedule[day] = moviesSchedule[day] || [];
    moviesSchedule[day].push({ title, timing });
    fs.writeFileSync('./data/schedule.json', JSON.stringify(moviesSchedule, null, 2));
    return JSON.stringify({ success: true, message: 'Movie added successfully.' });
  }
}

module.exports = Movie;
```

## Views:

- admin.ejs

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Cinemax</title>
  <link href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css"
rel="stylesheet">
  <link rel="stylesheet" href="css/style.css" type="text/css">
  <link rel="shortcut icon" href="/css/clapperboard.png" type="image/x-icon">
</head>

<body>
  <header>
    <%- include('navbar.ejs', { req: req }) %>
  </header>
  <main class="container">
    <h2 style="text-align: center; margin: 10px;">Add Movie</h2>
    <hr>
    <form action="/admin/add" method="post">
      <div class="grid-container">
        <div class="upload-preview">
          <label for="poster">Upload Poster:</label>
          <input type="file" id="poster" name="poster" accept="image/*">
          <div id="image-preview"></div>
        </div>

        <div class="add-movie-form">
          <div class="card">
            <div class="card-body">
              <div class="form-group">
                <label for="title">Title:</label>
                <input type="text" class="form-control" id="title" name="title"
required>
              </div>
              <div class="form-group">
                <label for="title">Storyline:</label>
                <textarea name="storyline" id="storyline" cols="50"
rows="5"></textarea>
            </div>
          </div>
        </div>
      </div>
    </form>
  </main>
</body>
</html>
```

```

        </div>
        <div class="form-group">
            <label for="timing">Timing:</label>
            <input type="time" class="form-control" id="timing"
name="timing" required>
        </div>
        <div class="form-group">
            <label for="cast">Cast:</label>
            <input type="text" class="form-control" id="cast" name="cast"
required>
        </div>
        <div class="form-group">
            <label for="day">Day:</label>
            <select class="form-control" id="day" name="day" required>
                <option value="Monday">Monday</option>
                <option value="Tuesday">Tuesday</option>
                <option value="Wednesday">Wednesday</option>
                <option value="Thursday">Thursday</option>
                <option value="Friday">Friday</option>
                <option value="Saturday">Saturday</option>
                <option value="Sunday">Sunday</option>
            </select>
        </div>
        <button type="submit" class="btn btn-primary">Add Movie</button>
    </div>
</div>
</div>
</div>
</form>
</main>
<%- include('footer.ejs') %>
    <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>
    <script
src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.5.4/dist/umd/popper.min.js"></scr
ipt>
    <script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
</body>

</html>

<script>
    document.getElementById('poster').addEventListener('change', function () {

```

```

var file = this.files[0];
var reader = new FileReader();

reader.onload = function (e) {
    var imagePreview = document.getElementById('image-preview');
    imagePreview.innerHTML = ""; // Clear previous preview

    var img = new Image();
    img.src = e.target.result;
    img.style.maxWidth = '756px'; // Adjust max-width as needed
    imagePreview.appendChild(img);
};

reader.readAsDataURL(file);
});
</script>

<script>
document.querySelector('form').addEventListener('submit', function (event) {
    event.preventDefault(); // Prevent default form submission

    // Get form data
    var addMovieFormData = new FormData();
    var fileInput = document.getElementById('poster').files[0];
    addMovieFormData.append('poster', fileInput); // Append the file to form data

    // Get other form fields
    var title = document.getElementById('title').value;
    var storyline = document.getElementById('storyline').value;
    var timing = document.getElementById('timing').value;
    var cast = document.getElementById('cast').value;
    var day = document.getElementById('day').value;

    console.log(title, timing, cast, day)

    addMovieFormData.append('title', title);
    addMovieFormData.append('storyline', storyline);
    addMovieFormData.append('timing', timing);
    addMovieFormData.append('cast', cast);
    addMovieFormData.append('day', day);

    // Send POST request to server
    fetch('/admin/add', {

```

```

        method: 'POST',
        body: addMovieFormData
    })
    .then(response => response.json())
    .then(data => {
        console.log('Response from server:', data);
        window.location.href = "/";
        // Handle response from server
    })
    .catch(error => {
        console.error('Error:', error);
        // Handle error
    });
});
</script>

```

- footer.ejs

```

<footer class="footer">
  <div class="container">
    <span class="text-muted">&copy; 2024 Cinemax</span>
  </div>
</footer>

```

- Index.ejs

```

<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Cinemax</title>
  <link href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css"
rel="stylesheet">
  <link rel="stylesheet" href="css/style.css" type="text/css">
  <link rel="shortcut icon" href="/css/clapperboard.png" type="image/x-icon">
</head>

<body>
  <header>

```



```

    <%- include('navbar.ejs', { req: req }) %>
</header>
<div class="container">
  <h2>Current Movies</h2>
  <div class="row">
    <% currentMovies.forEach(movie => { %>
      <%- include('movieCard.ejs', { movie }) %>
    <% }); %>
  </div>
  <h2>Upcoming Movies</h2>
  <div class="row">
    <% upcomingMovies.forEach(movie => { %>
      <%- include('movieCard.ejs', { movie }) %>
    <% }); %>
  </div>
</div>
  <%- include('footer.ejs') %>
<script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
</body>
</html>

```

#### • Login.ejs

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Login - Cinemax</title>
  <!-- Link to Bootstrap CSS -->
  <link href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css"
rel="stylesheet">
  <!-- Link to the CSS file -->
  <link rel="stylesheet" href="/css/style.css">
  <link rel="shortcut icon" href="/css/clapperboard.png" type="image/x-icon">
</head>
<body>
  <header>
    <%- include('navbar.ejs', { req: req }) %>

  </header>

```

```

<div class="container authForms">
  <form class="login-form" action="/auth/login" method="post">
    <div class="card ">
      <div class="card-body">
        <h1 class="text-center mb-4">Login</h1>
        <div class="form-group">
          <input type="text" class="form-control" id="username"
name="username" placeholder="Username" required>
        </div>
        <div class="form-group">
          <input type="password" class="form-control" id="password"
name="password" placeholder="Password"required>
        </div>
        <button type="submit" class="btn btn-primary">Login</button>
        <p class="message">Not registered? <a href="/auth/signup">Create an
account</a></p>
      </div>
    </div>
  </form>
</div>
<%- include('footer.ejs') %>
<!-- Bootstrap JS -->
<script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
</body>
</html>

```

- movieCard.ejs

```

<div class="col-md-3 mb-4">
  <div class="card">
    <div class="card-img-container">
      ">
    <div class="overlay">
      <button class="btn btn-info infoButton"
onclick="showMovieInfo('<%=movie.cast%>')">Info</button>
    <div class="movie-info hidden">
      Cast: <%= movie.cast %>
      <hr>
      Storyline: <%= movie.storyline %>
    </div>
  </div>

```

```

        </div>
        <div class="movie-title">
            <h5 class="card-title">
                <%= movie.title %>
            </h5>
        </div>
    </div>
</div>

<script>
    document.addEventListener("DOMContentLoaded", function () {
        // Get all movie cards
        var movieCards = document.querySelectorAll('.card');

        // Add event listeners for each movie card
        movieCards.forEach(function (card) {
            var overlay = card.querySelector('.overlay');
            var movieInfoElement = card.querySelector('.movie-info');
            var infoButton = overlay.querySelector('.infoButton');

            // Hide cast when mouse leaves the card
            overlay.addEventListener('mouseleave', function () {
                movieInfoElement.classList.add('hidden');
                infoButton.classList.remove('hidden');
            });

            // Show cast when the info button is clicked
            infoButton.addEventListener('click', function () {
                movieInfoElement.classList.remove('hidden');
                infoButton.classList.add('hidden');
            });
        });
    });
</script>

```

• navbar.ejs

```
<nav class="navbar navbar-expand navbar-light">
  <a class="navbar-brand" href="/" style="display: flex; align-items: center;">
    <script src="https://cdn.lordicon.com/lordicon.js"></script>
    <lord-icon src="https://cdn.lordicon.com/tlvdupnp.json" trigger="hover"
style="width: 75px; height: 75px;"></lord-icon>
    <h2 id="LogoText">Cinemax</h2>
  </a>
  <div class="navbar-collapse collapse justify-content-end" id="navbarNav">
    <ul class="navbar-nav">
      <li class="nav-item">
        <button class="btn btn-warning" style="margin-right: 2px;"
onclick="window.location.href='/schedule'">Schedule</button>
      </li>
      <% if (req.session.authenticated) { %>
        <li class="nav-item">
          <button class="btn btn-success" style="margin-right: 2px;"
onclick="window.location.href='/admin'">Add Movie</button>
        </li>
        <li class="nav-item">
          <button class="btn btn-primary" style="margin-right: 2px;"
onclick="window.location.href='/auth/logout'">Logout</button>
        </li>
        <% } else if (!req.session.authenticated) { %>
          <li class="nav-item">
            <button class="btn btn-primary" style="margin-right: 2px;"
onclick="window.location.href='/auth/login'">Login</button>
          </li>
          <li class="nav-item">
            <button class="btn btn-primary" style="margin-right: 2px;"
onclick="window.location.href='/auth/signup'">Signup</button>
          </li>
          <% } %>
        </ul>
      </div>
    </nav>
```

• registration.ejs

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Registration - Cinemax </title>
  <link href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css"
rel="stylesheet">
  <link rel="stylesheet" href="/css/style.css">
  <link rel="shortcut icon" href="/css/clapperboard.png" type="image/x-icon">
</head>

<body>
  <header>
    <%- include('navbar.ejs', { req: req }) %>
  </header>
  <div class="container authForms">
    <form class="register-form" action="/auth/signup" method="POST">
      <div class="card ">
        <div class="card-body">
          <h1 class="text-center mb-4">Registration</h1>
          <div class="form-group">
            <input type="text" class="form-control" id="fullName"
name="fullName" placeholder="Full Name" required>
          </div>
          <div class="form-group">
            <input type="text" class="form-control" id="username"
name="username" placeholder="Username" required>
          </div>
          <div class="form-group">
            <input type="password" class="form-control" id="password"
name="password" placeholder="Password"required>
          </div>
          <button type="submit" class="btn btn-primary">Register</button>
          <p class="message">Already registered? <a href="/auth/login">Login
here</a></p>
        </div>
      </div>
    </form>
  </div>
```

```

    <%- include('footer.ejs') %>

    <script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
  </body>
</html>

```

- schedule.ejs

```

<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Schedule - Cinemax</title>
  <link href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css"
rel="stylesheet">
  <!-- Link to the CSS file -->
  <link rel="stylesheet" href="/css/style.css">
  <link rel="shortcut icon" href="/css/clapperboard.png" type="image/x-icon">
</head>

<body>
  <header>
    <%- include('navbar.ejs', { req: req }) %>
  </header>
  <main class="container">
    <h2>Weekly Schedule</h2>
    <div class="card">
      <div class="card-content">
        <% Object.keys(schedule).forEach(day=> { %>
          <hr>
          <h3 style="text-align: center;">
            <%= day %>
          </h3>
          <hr>
          <% if (schedule[day].length> 0) { %>
            <ul>
              <% const sortedMovies=schedule[day].sort((a, b)=> {
                // Assuming timing is in the format 'HH:MM AM/PM'
                const timeA = a.timing.split(' ')[0];

```

```

const timeB = b.timing.split(' ')[0];
return new Date('1970/01/01 ' + timeA) - new Date('1970/01/01 ' +
timeB);

});
sortedMovies.forEach(movie => { %>
<li style="text-align: center;">
    <%= movie.title %> - <%= movie.timing %>
</li>
<% }); %>
</ul>
<% } else { %>
    <p style="text-align: center;">No movies scheduled for <%= day
%>.</p>
    <% } %>
    <% }); %>
<hr>
</div>
</div>
</main>

<%- include('footer.ejs') %>
<script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
</body>

</html>

```

- Style.css

```
* {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
    font-family: 'Trebuchet MS', 'Lucida Sans Unicode', 'Lucida Grande', 'Lucida Sans',
    Arial, sans-serif;
}

/* Body styles */
body {
    font-family: Arial, sans-serif;
    line-height: 1.6;
    background-color: #f8f9fa;
    color: #333;
}

/* Header styles */
header {
    background: linear-gradient(to right, #ff7e5f, #fc720a);
    color: #fff;
    text-align: center;
}

/* Navbar styles */
nav {
    display: flex;
    justify-content: center;
}

nav a {
    color: #fff;
    text-decoration: none;
    margin: 0 10px;
}

/* Container styles */
.container {
    max-width: 1200px;
    margin: 20px auto;
    padding: 0 20px;
}
```



```
/* Card styles */
.card {
  background: linear-gradient(to right, #d4fc79, #96e6a1);
  border-radius: 10px;
  box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
  margin-bottom: 20px;
  position: relative;
}

.card h2 {
  padding: 15px;
  border-bottom: 1px solid rgba(0, 0, 0, 0.1);
}

.card-content {
  padding: 20px;
}

/* List styles */
ul {
  list-style: none;
}

/* Movie title grid */
.card-content ul {
  display: grid;
  grid-template-columns: repeat(auto-fit, minmax(200px, 1fr));
  grid-gap: 10px;
}

/* Footer styles */
footer {
  background: linear-gradient(to right, #45a247, #283c86);
  color: #fff;
  text-align: center;
  padding: 20px 0;
}

.card-img-container {
  position: relative;
  overflow: hidden;
  border-radius: 10px;
```

```
}
```

```
.card-img-container .overlay {  
  display: flex;  
  flex-direction: column;  
  justify-content: center;  
  align-items: center;  
  position: absolute;  
  top: 0;  
  left: 0;  
  width: 100%;  
  height: 100%;  
  background-color: rgba(0, 0, 0, 0.5);  
  opacity: 0;  
  transition: opacity 0.3s ease;  
}
```

```
.card-img-container:hover .overlay {  
  opacity: 1;  
}
```

```
.card-img-container:hover img {  
  filter: brightness(70%);  
}
```

```
.movie-title {  
  position: absolute;  
  top: 10px;  
  left: 10px;  
  color: white;  
  margin: 0;  
  padding: 0;  
  z-index: 1;  
}
```

```
.movie-info {  
  z-index: 2;  
}
```

```
.authForms {  
  max-width: 50%;  
}
```

```
.customNavbar {
  display: grid;
  grid-template-columns: 1fr 2fr;
  grid-template-rows: 1fr;
  gap: 0px 10px;
  grid-auto-flow: row;
  justify-items: start;
  grid-template-areas:
    "Logo NavItems";
}
```

```
.Logo {
  grid-area: Logo;
}
```

```
.NavItems {
  grid-area: NavItems;
  justify-items: end;
  justify-content: end;
}
```

```
#LogoText {

  display: inline-block;
  margin-left: 10px;
  vertical-align: middle;
  font-family: 'Trebuchet MS', 'Lucida Sans Unicode', 'Lucida Grande', 'Lucida Sans',
  Arial, sans-serif;
  text-transform: uppercase;
}
```

```
.grid-container {
  display: grid;
  grid-template-columns: repeat(2, 1fr);
  gap: 20px;
}
```

```
.hidden {
  display: none;
}
```

```
.info-button {
  /* New styles for the "Info" button */
}
```

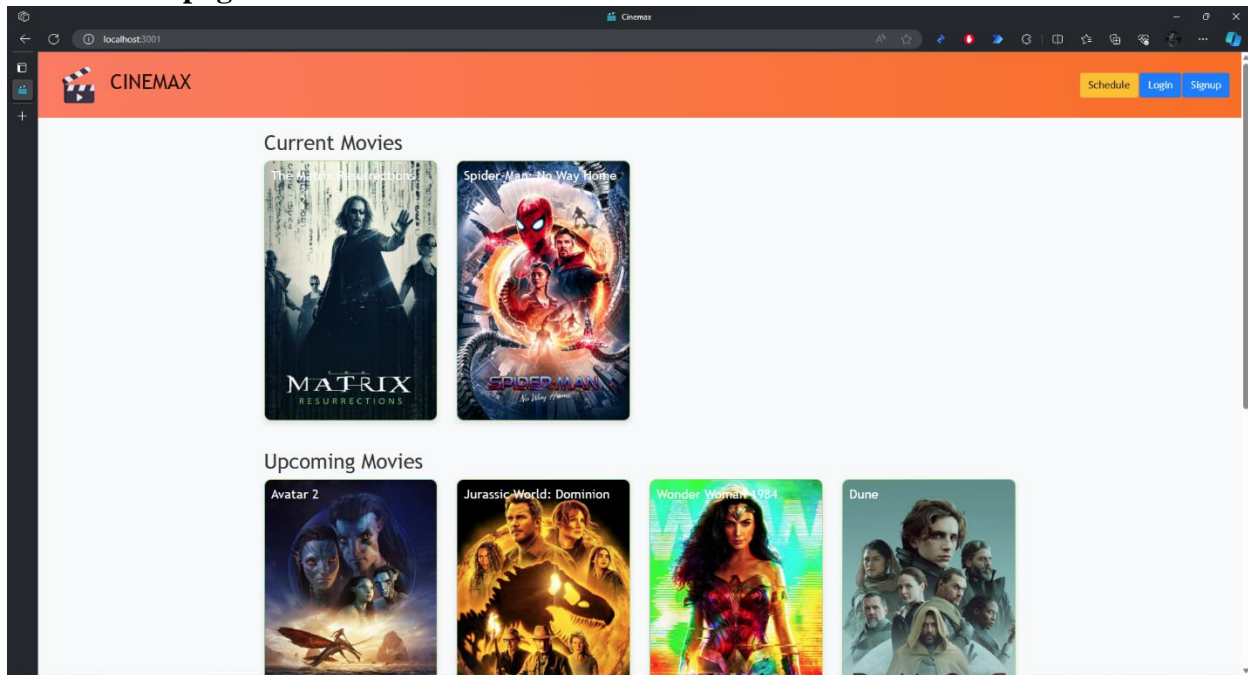
```
position: absolute;
top: 50%;
left: 50%;
transform: translate(-50%, -50%);
z-index: 2;
}
```

```
.movie-info {
  /* New styles to display cast and storyline below each other */
  color: white;
  text-align: start;
  padding: 10px;
  position: absolute;
  bottom: 0;
  left: 0;
  width: 100%;
  background-color: rgba(0, 0, 0, 0.7);
  opacity: 0;
  transition: opacity 0.3s ease;
  z-index: 1;
}
```

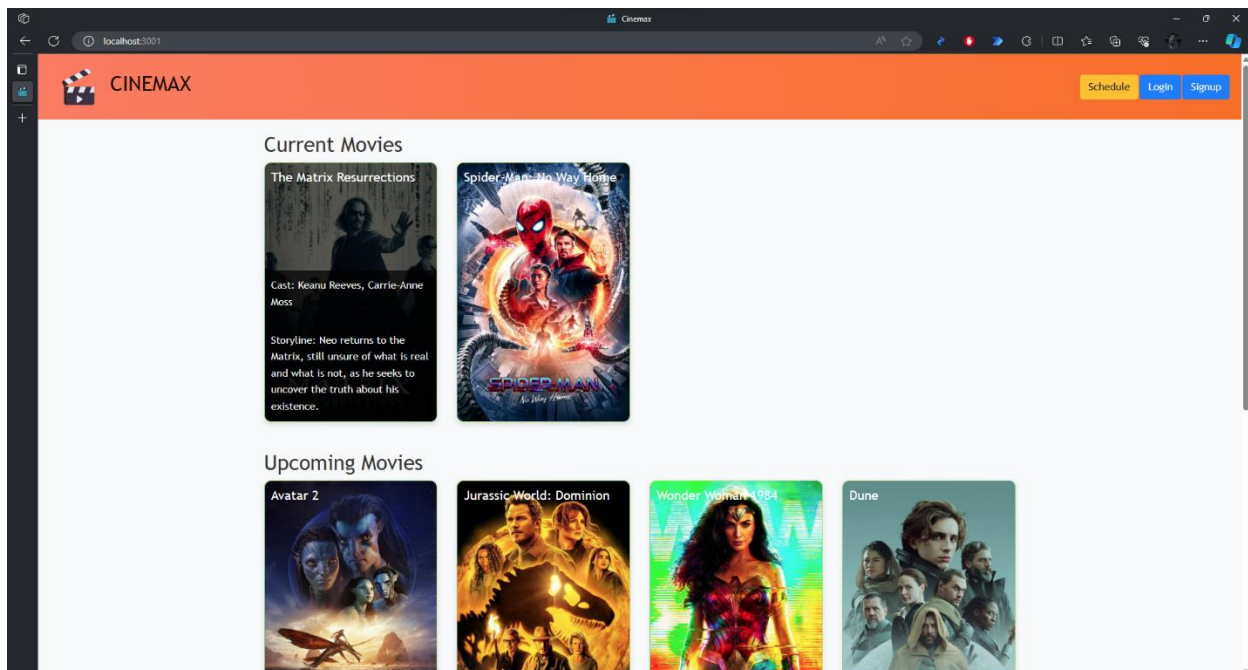
```
.card:hover .movie-info {
  /* New styles to display cast and storyline when hovering over the card */
  opacity: 1;
}
```

Output:

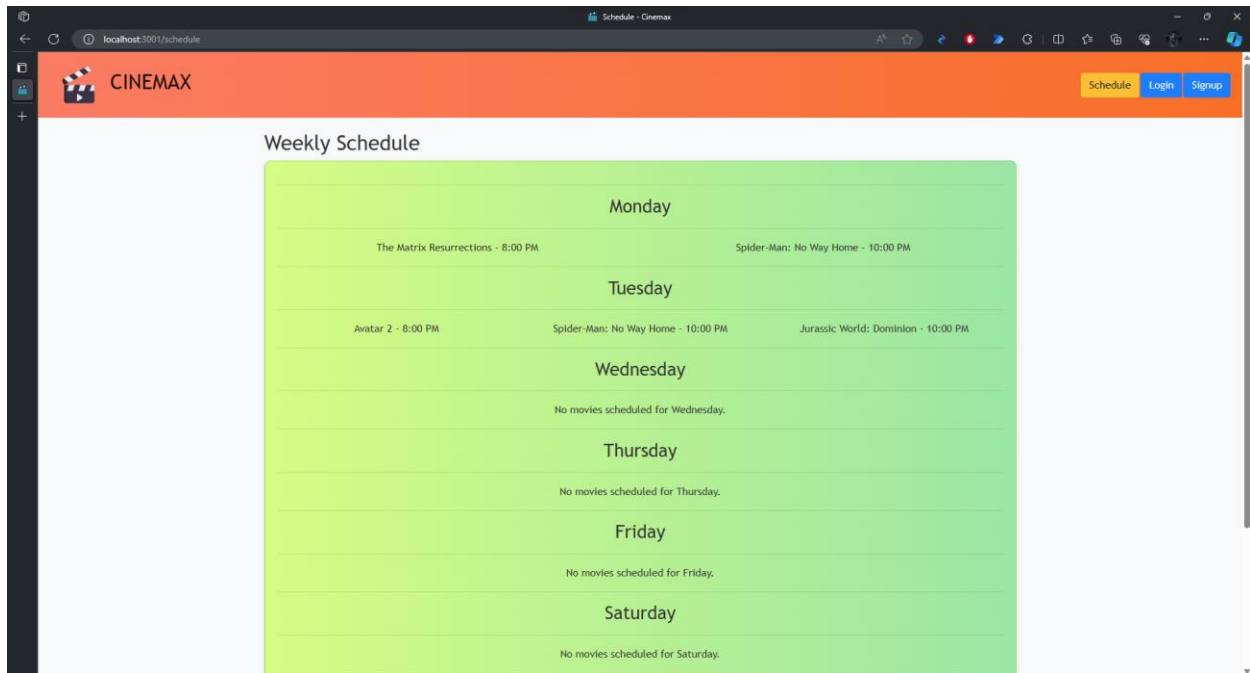
- Homepage



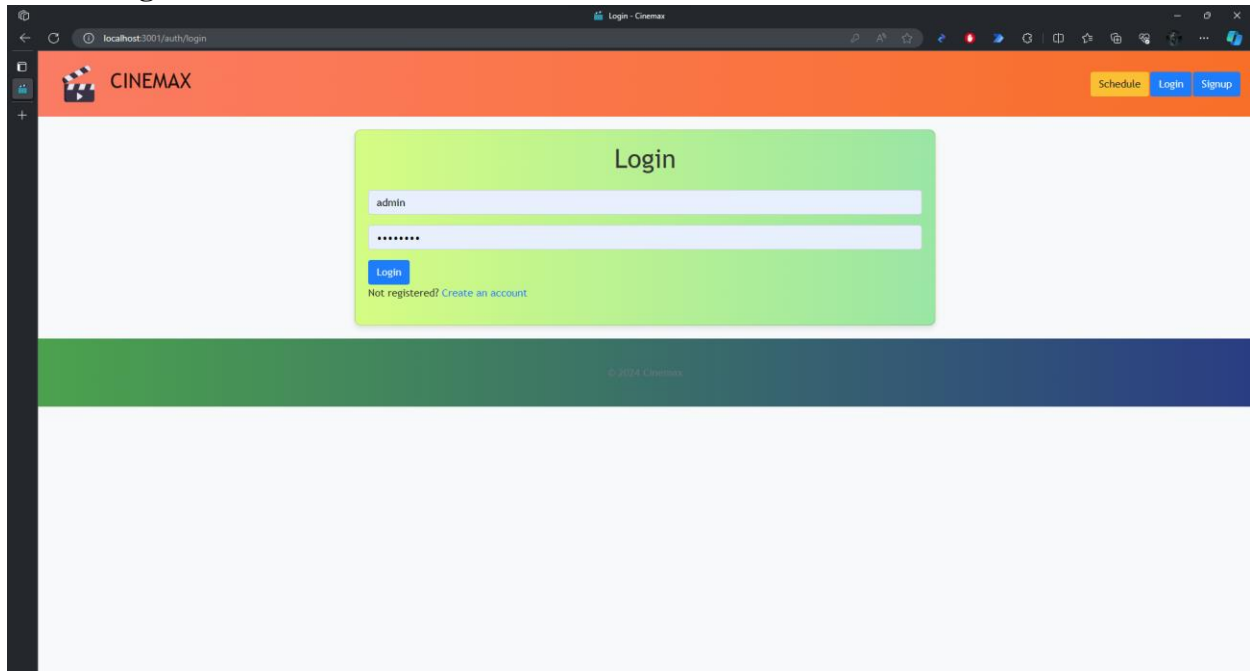
- Movie Information



- Schedule



- Login



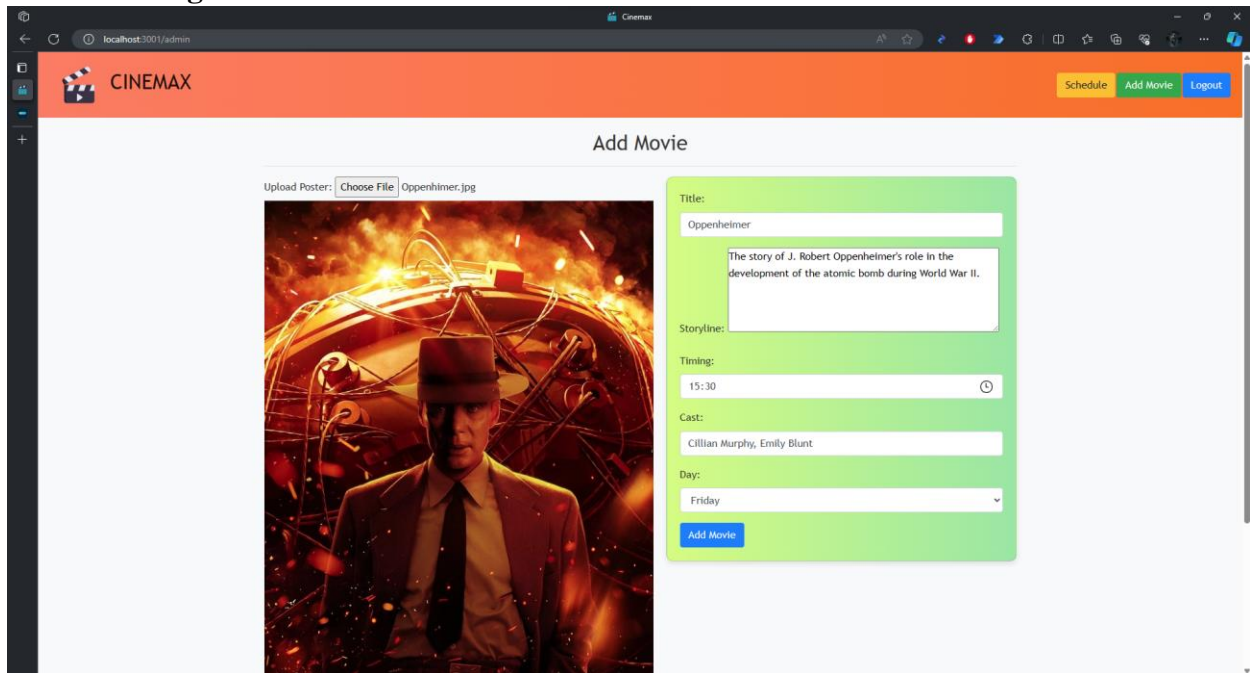
- Registration

The screenshot shows the 'Registration' page of the CINEMAX application. The browser address bar indicates the URL is 'localhost:3001/auth/signup'. The page features a dark sidebar on the left with a clapperboard icon and a '+' sign. The top navigation bar is orange, displaying the 'CINEMAX' logo and buttons for 'Schedule', 'Login', and 'Signup'. The main content area has a light gray background with a central green registration form. The form includes input fields for 'Full Name', 'Username', and 'Password', a 'Register' button, and a link for 'Already registered? Login here'. A footer bar at the bottom contains a copyright notice: '© 2024 L. Cinemax'.

- Add Movie

The screenshot displays the 'Add Movie' page in the CINEMAX application. The browser address bar shows 'localhost:3001/admin'. The interface includes a dark sidebar with a clapperboard icon and a '+' sign. The top navigation bar is orange, with the 'CINEMAX' logo and buttons for 'Schedule', 'Add Movie', and 'Logout'. The main content area has a light gray background with a central green form for adding a movie. The form contains fields for 'Title', 'Storyline', 'Timing' (with a clock icon), 'Cast', and 'Day' (a dropdown menu currently showing 'Monday'). An 'Add Movie' button is at the bottom of the form. To the left of the form, there is a section for 'Upload Poster' with a 'Choose File' button and the text 'No file chosen'. A footer bar at the bottom displays the copyright notice: '© 2024 L. Cinemax'.

- Adding Movie



- Movie Added

