# Full Stack Movie List Application

This project is a full-stack web application that includes user registration, login, and a movie list feature. The backend is built using Node.js and MongoDB, and the frontend is developed using Vue.js.

## Features

- User Registration

- User Login

- Home Page

- Registered Users Page

- Company Info Page

- Movie List Page

## Prerequisites

- Node.js

- MongoDB

- Vue CLI

## Backend Setup

### 1. Install Dependencies

Navigate to the `backend` directory and install the required dependencies:

```sh

cd backend

npm install

```

### 2. Configure Environment Variables

Create a `.env` file in the `backend` directory and add your MongoDB connection string and other environment variables:

```sh

MONGODB\_URI=<your-mongodb-connection-string>

PORT=3000

JWT\_SECRET=<your-jwt-secret>

```

### 3. Run the Server

Start the backend server:

```sh

node server.js

```

The backend server should now be running on `http://localhost:3000`.

## Frontend Setup

### 1. Install Dependencies

Navigate to the `frontend` directory and install the required dependencies:

```sh

cd frontend

npm install

```

### 2. Update API Endpoints

Ensure the API endpoints in your frontend code point to your backend server. Update any necessary configuration files if needed.

### 3. Run the Development Server

Start the frontend development server:

```sh

npm run serve

```

The frontend application should now be running on `http://localhost:8080`.

## Project Structure

### Backend

- `server.js`: Entry point for the backend server.

- `models/`: Contains MongoDB models.

- `routes/`: Contains API routes.

- `controllers/`: Contains route handlers.

### Frontend

- `src/`: Contains the main source code for the Vue.js application.

- `components/`: Contains Vue components.

- `router/`: Contains route definitions.

- `views/`: Contains view components.

## Vue Components

### HomePage.vue

```vue

<template>

<div>

<h2>Home Page</h2>

<nav>

<router-link to="/login">Login</router-link>

<router-link to="/signup">Signup</router-link>

</nav>

<router-view></router-view>

</div>

</template>

<script>

export default {

name: 'HomePage'

};

</script>

```

### UserLogin.vue

```vue

<template>

<div>

<h2>Login Page</h2>

</div>

</template>

<script>

export default {

name: 'UserLogin'

};

</script>

```

### UserSignup.vue

```vue

<template>

<div>

<h2>Signup Page</h2>

</div>

</template>

<script>

export default {

name: 'UserSignup'

};

</script>

```

### router/index.js

```js

import { createRouter, createWebHistory } from 'vue-router';

import HomePage from '../components/HomePage.vue';

import UserLogin from '../components/UserLogin.vue';

import UserSignup from '../components/UserSignup.vue';

const routes = [

{ path: '/', component: HomePage },

{ path: '/login', component: UserLogin },

{ path: '/signup', component: UserSignup },

{ path: '/:pathMatch(.\*)\*', redirect: '/login' }

];

const router = createRouter({

history: createWebHistory(),

routes

});

export default router;

```

### main.js

```js

import { createApp } from 'vue';

import App from './App.vue';

import router from './router';

createApp(App).use(router).mount('#app');

```

## Additional Notes

- Ensure MongoDB is running and accessible.

- Make sure to replace placeholder values in the `.env` file with actual values.

- Update any API endpoints in the frontend to match the backend server's URL.

## License

This project is licensed under the MIT License.

---