**How to Do setup & all >>>>>**

Step 1: Set up the Vue.js Project >>>

First, create a new Vue project if you haven't already. You can use Vue CLI for this.



-----------------------------------------------------------

Step 2: Set up the Folder Structure

The following structure gives an organized view of how components, services, and store modules will be used.

restaurant-app/

│

├── public/

├── src/

│ ├── assets/ # Store static assets

│ ├── components/ # Reusable components

│ ├── Home.vue # Home page

│ ├── AddRestaurant.vue # Add Restaurant page

│ ├── UpdateRestaurant.vue # Update Restaurant page

│ ├── Login.vue # Login page

│ ├── CreateUser.vue # Create new user page

│ ├── router/ # Vue Router

│ └── index.js # Route configuration

│ ├── store/ # Vuex store

│ └── index.js # Vuex store configuration

│ ├── services/ # API services

│ └── authService.js # Service for handling authentication

│ ├── App.vue # Root component

│ ├── main.js # Entry file

│

└── package.json

-----------------------------------------------------------

Step 3: Install Dependencies

You’ll need a few dependencies for basic functionality like form handling, Vue Router, Vuex, and making API calls.

# npm install axios

services/authService.js

-----------------------------------------------------------

Step 4: Define API Service

Create an authService.js inside the services/ folder to handle authentication API calls.

import axios from 'axios';

const API\_URL = '<http://localhost:8000/api>'; // Replace with actual backend API URL

export default {

login(email, password) {

return axios.post(`${API\_URL}/login`, { email, password });

},

register(username, email, password) {

return axios.post(`${API\_URL}/register`, { username, email, password });

}

};

-----------------------------------------------------------

Step 5: Set up Vuex Store

Use Vuex for handling global state, including user authentication.

store/index.js

import Vue from 'vue';

import Vuex from 'vuex';

import authService from '../services/authService';

Vue.use(Vuex);

export default new Vuex.Store({

state: {

user: null, // Store logged-in user data

token: null,

},

mutations: {

SET\_USER(state, user) {

state.user = user;

},

SET\_TOKEN(state, token) {

state.token = token;

},

LOGOUT(state) {

state.user = null;

state.token = null;

}

},

actions: {

login({ commit }, { email, password }) {

return authService.login(email, password).then(response => {

commit('SET\_USER', response.data.user);

commit('SET\_TOKEN', response.data.token);

localStorage.setItem('token', response.data.token); // Save token in localStorage

});

},

logout({ commit }) {

commit('LOGOUT');

localStorage.removeItem('token');

},

register({ dispatch }, { username, email, password }) {

return authService.register(username, email, password).then(() => {

// Auto-login after registration

dispatch('login', { email, password });

});

}

}

});

----------------------------------------------------

Step 6: Set up Vue Router

Define routes for Home, Add Restaurant, Update Restaurant, Login, and Create User pages.

router/index.js

import Vue from 'vue';

import Router from 'vue-router';

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

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

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

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

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

Vue.use(Router);

export default new Router({

routes: [

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

{ path: '/add-restaurant', component: AddRestaurant },

{ path: '/update-restaurant/:id', component: UpdateRestaurant },

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

{ path: '/create-user', component: CreateUser }

]

});

Step 7: Create the Pages

Home.vue