# Практика 4

Для начала необходимо создать новую папку (**НАЗОВИТЕ ПАПКУ vue**), в которой будут располагаться весь frontend

Создать её можно рядом с папкой backend-a проекта. Название неважно.

# Шаг 1: Установка Vue CLI

В папке проекта необходимо открыть терминал и ввести следующую команду

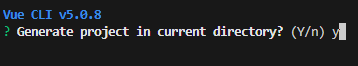
npm install -g @vue/cli

# Шаг 2: Создание нового проекта Vue

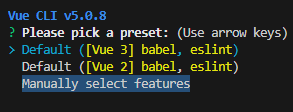
1. Создайте новый проект Vue с помощью Vue CLI:

vue create .

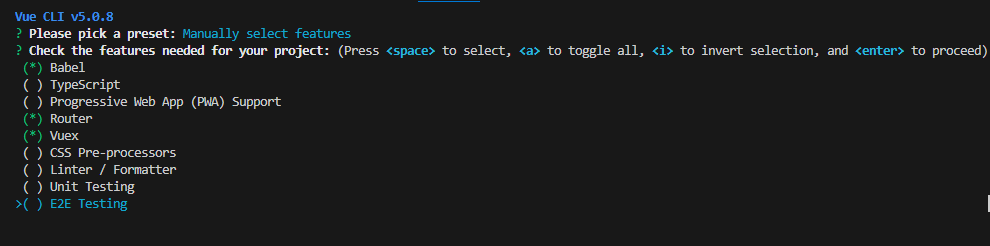
После чего появиться следующий вопрос, на который нужно ответить y (игрик)



1. Далее необходимо выбрать третий вариант установки



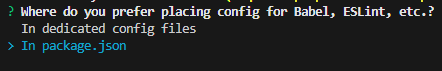
1. С помощью **пробела** выбираем следующие поля и нажимаем **enter**



1. В следующем поле выбираем третью версию и после появиться сообщение, на которое необходимо ответить y



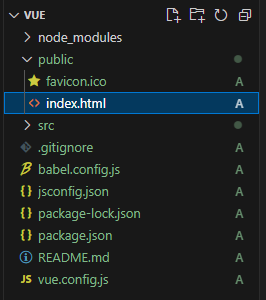
1. Далее выбираем «In package.json»



1. Установите Axios для выполнения HTTP-запросов к вашему серверу:

npm install axios

1. Далее файл index.html:



меняем на:

<!DOCTYPE html>

<html lang="">

  <head>

    <meta charset="utf-8">

    <meta name="viewport" content="width=device-width,initial-scale=1.0">

    <title>Записываемся на ноготочки</title>

  </head>

  <body>

    <div id="app">

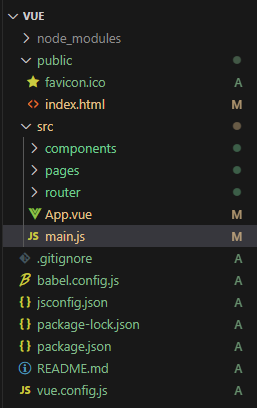
    </div>

  </body>

</html>

1. Далее в папке **src** можно удалить **assets, store, views и создать папку pages**

В итоге у вас должно остаться следующая структура



# Шаг 3:

1. В файле main.js вписываем следующую структуру

import { createApp } from 'vue'

import App from './App.vue'

import router from './router/router'

const app = createApp(App)

createApp(App).mount('#app')

app

    .use(router)

    .mount('#app');

1. В файле App.vue оставляем следующую структуру

<template>

  <div class="app">

    <router-view></router-view>

  </div>

</template>

<script>

</script>

<style>

</style>

В папке **pages** создаем файл **Main.vue** и в нем создаем следующую структуру:

<template>

  <div class="main-container">

    <h1>Добро пожаловать в систему бронирования "Записываемся на ноготочки"</h1>

    <p>Здесь вы можете записаться к мастерам маникюра и педикюра</p>

    <router-link to="/register" class="btn">Регистрация</router-link>

    <router-link to="/login" class="btn">Вход</router-link>

    <router-link to="/admin-login" class="btn">Админ-панель</router-link>

  </div>

</template>

<script>

export default {

  name: 'Main'

}

</script>

<style scoped>

.main-container {

  max-width: 600px;

  margin: 0 auto;

  text-align: center;

}

.btn {

  display: inline-block;

  margin: 10px;

  padding: 10px 20px;

  background-color: #007bff;

  color: #fff;

  text-decoration: none;

  border-radius: 5px;

  cursor: pointer;

}

.btn:hover {

  background-color: #0056b3;

}

</style>

# Шаг 4: Реализация компонентов Vue

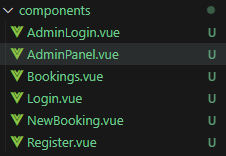
Создание компонентов:

Удалите файл components/HelloWorld.vue который является частью стандартного шаблона Vue CLI.

Вам нужно создать компоненты Vue для каждой страницы вашего приложения. В вашем случае, вам потребуются компоненты для страницы регистрации, авторизации, просмотра заявлений и панели администратора. Создайте файлы .vue для каждой страницы, например **Login.vue,** **Bookings.vue, Register.vue, NewBooking.vue, AdminPanel.vue и**

**Эти компоненты создаются в папке src/components**

В папке components должна получиться следующая структура:



Каждый файл отражает страницу, которая отвечает на необходимые ендпоинты, определённые на сервере.

Далее необходимо заполнить каждый файл

**Первый компонент – страница регистрации (Register.vue)**

<template>

    <div class="register-container">

      <h2>Регистрация</h2>

      <form @submit.prevent="registerUser">

        <label for="fullName">ФИО:</label>

        <input type="text" id="full\_name" v-model="full\_name" required>

        <label for="phone">Телефон:</label>

        <input type="tel" id="phone" v-model="phone" required>

        <label for="login">Логин:</label>

        <input type="text" id="login" v-model="login" required>

        <label for="password">Пароль:</label>

        <input type="password" id="password" v-model="password" required>

        <button type="submit">Зарегистрироваться</button>

      </form>

      <p>Уже есть аккаунт? <router-link to="/login">Войдите</router-link></p>

    </div>

  </template>

  <script>

  import axios from 'axios'; // импортируем axios

  export default {

    data() {

      return {

        full\_name: '',

        phone: '',

        login: '',

        password: '',

        message: ''

      };

    },

    methods: {

      async registerUser() { // изменяем название метода на registerUser

        try {

          const response = await axios.post('http://localhost:8000/register', {

            full\_name: this.full\_name,

            phone: this.phone,

            login: this.login,

            password: this.password,

          });

          if (response.status === 201) {

            this.message = 'Регистрация успешна';

          } else {

            this.message = response.data.error;

          }

        } catch (error) {

          this.message = 'Ошибка регистрации';

        }

      },

    },

  };

  </script>

  <style scoped>

  .register-container {

    max-width: 400px;

    margin: 0 auto;

  }

  label {

    display: block;

    margin-bottom: 5px;

  }

  input[type="text"],

  input[type="tel"],

  input[type="password"] {

    width: 100%;

    padding: 8px;

    margin-bottom: 10px;

  }

  button {

    background-color: #007bff;

    color: #fff;

    border: none;

    padding: 10px 20px;

    border-radius: 5px;

    cursor: pointer;

  }

  button:hover {

    background-color: #0056b3;

  }

  p {

    margin-top: 20px;

    text-align: center;

  }

  p a {

    color: #007bff;

    text-decoration: none;

  }

  p a:hover {

    text-decoration: underline;

  }

  </style>

# Компонент для страницы входа (Login.vue):

<template>

    <div class="login">

      <h2>Авторизация</h2>

      <form @submit.prevent="loginUser">

        <div>

          <label>Логин</label>

          <input v-model="login" type="text" required />

        </div>

        <div>

          <label>Пароль</label>

          <input v-model="password" type="password" required />

        </div>

        <button type="submit">Войти</button>

      </form>

      <p v-if="message">{{ message }}</p>

      <router-link to="/">На главную</router-link>

    </div>

</template>

<script>

import axios from 'axios';

export default {

    data() {

        return {

            login: '',

            password: '',

            message: ''

        };

    },

    methods: {

        async loginUser() {

            try {

                const response = await axios.post('http://localhost:8000/login', {

                    login: this.login,

                    password: this.password

                }, {

                    withCredentials: true // Отправка cookies с запросом

                });

                if (response.data.message === 'Login successful') {

                    this.$router.push('/bookings');

                } else {

                    this.message = response.data.error;

                }

            } catch (error) {

                this.message = 'Ошибка авторизации';

            }

        }

    }

};

</script>

<style scoped>

.login {

    max-width: 400px;

    margin: 0 auto;

}

label {

    display: block;

    margin-bottom: 5px;

}

input[type="text"],

input[type="password"] {

    width: 100%;

    padding: 8px;

    margin-bottom: 10px;

}

button {

    background-color: #007bff;

    color: #fff;

    border: none;

    padding: 10px 20px;

    border-radius: 5px;

    cursor: pointer;

}

button:hover {

    background-color: #0056b3;

}

p {

    margin-top: 20px;

    text-align: center;

}

</style>

# Компонент для страницы просмотра заявлений пользователя (Bookings.vue):

<template>

    <div class="bookings">

        <h2>Мои заявки</h2>

        <div v-if="bookings.length">

            <div v-for="booking in bookings" :key="booking.id" class="booking">

                <p>Мастер: {{ booking.master\_name }}</p>

                <p>Дата и время: {{ booking.booking\_datetime }}</p>

                <p>Статус: {{ booking.status\_name }}</p>

            </div>

        </div>

        <p v-else>Заявки отсутствуют.</p>

        <router-link to="/new-booking" class="link">Создать новую заявку</router-link>

        <router-link to="/" class="link">Главное меню</router-link>

    </div>

</template>

<script>

import axios from 'axios';

export default {

    data() {

        return {

            bookings: [],

        };

    },

    async created() {

        try {

            const response = await axios.get('http://localhost:8000/bookings', {

                withCredentials: true

            });

            this.bookings = response.data;

        } catch (error) {

            console.error('Ошибка при загрузке заявок:', error);

            if (error.response && error.response.status === 401) {

                this.$router.push('/login');

            }

        }

    },

};

</script>

<style scoped>

.bookings {

    max-width: 600px;

    margin: 0 auto;

}

.booking {

    border: 1px solid #ccc;

    padding: 15px;

    margin-bottom: 15px;

    border-radius: 5px;

}

h2 {

    text-align: center;

    margin-bottom: 20px;

}

p {

    margin: 5px 0;

}

.link {

    display: inline-block;

    margin-top: 10px;

    background-color: #007bff;

    color: #fff;

    padding: 10px 20px;

    border-radius: 5px;

    text-decoration: none;

}

.link:hover {

    background-color: #0056b3;

}

</style>

# Компонент для добавления новой заявки (NewBooking.vue):

<template>

  <div class="new-booking">

    <h2>Новая заявка</h2>

    <!-- Форма для создания новой заявки -->

    <form @submit.prevent="createBooking">

      <!-- Выбор мастера -->

      <div>

        <label>Мастер</label>

        <select v-model="id\_master" required>

          <option v-for="master in masters" :key="master.id" :value="master.id">{{ master.name }}</option>

        </select>

      </div>

      <!-- Ввод даты и времени -->

      <div>

        <label>Дата и время</label>

        <input v-model="booking\_datetime" type="datetime-local" required />

      </div>

      <!-- Кнопка отправки заявки -->

      <button type="submit">Отправить заявку</button>

    </form>

    <!-- Сообщение об успешном создании заявки или об ошибке -->

    <p v-if="message">{{ message }}</p>

    <!-- Кнопка перехода на страницу просмотра своих заявок -->

    <router-link to="/bookings">Мои заявки</router-link>

  </div>

</template>

<script>

import axios from 'axios';

export default {

  data() {

    return {

      id\_master: '',

      booking\_datetime: '',

      message: '',

      masters: [],

    };

  },

  async created() {

    try {

      const response = await axios.get('http://localhost:8000/masters', { withCredentials: true });

      if (response.status === 200) {

        this.masters = response.data;

      }

    } catch (error) {

      console.error('Ошибка при загрузке списка мастеров:', error);

    }

  },

  methods: {

    async createBooking() {

      try {

        const response = await axios.post('http://localhost:8000/bookings', {

          id\_master: this.id\_master,

          booking\_datetime: this.booking\_datetime,

        }, { withCredentials: true });

        if (response.status === 201) {

          this.message = 'Заявка успешно создана';

        } else {

          this.message = response.data.error;

        }

      } catch (error) {

        this.message = 'Ошибка при создании заявки';

      }

    },

  },

};

</script>

<style scoped>

.new-booking {

  max-width: 400px;

  margin: 0 auto;

}

form {

  margin-top: 20px;

}

label {

  display: block;

  margin-bottom: 5px;

}

select,

input[type="datetime-local"] {

  width: 100%;

  padding: 8px;

  margin-bottom: 10px;

}

button {

  background-color: #007bff;

  color: #fff;

  border: none;

  padding: 10px 20px;

  border-radius: 5px;

  cursor: pointer;

}

button:hover {

  background-color: #0056b3;

}

p {

  margin-top: 20px;

  text-align: center;

}

</style>

**Авторизация администратора**

<template>

  <div class="admin-login">

    <h2>Авторизация администратора</h2>

    <form @submit.prevent="adminLogin">

      <div>

        <label for="login">Логин:</label>

        <input type="text" id="login" v-model="login" required />

      </div>

      <div>

        <label for="password">Пароль:</label>

        <input type="password" id="password" v-model="password" required />

      </div>

      <button type="submit">Войти</button>

    </form>

  </div>

</template>

<script>

import axios from 'axios';

export default {

  data() {

    return {

      login: '',

      password: ''

    };

  },

  methods: {

    async adminLogin() {

      try {

        const response = await axios.post('http://localhost:8000/admin/login', {

          username: this.login,

          password: this.password

        }, { withCredentials: true });

        if (response.status === 200) {

          this.$router.push('/admin-panel'); // Перенаправляем на страницу панели администратора

        } else {

          console.error('Ошибка аутентификации');

        }

      } catch (error) {

        console.error('Ошибка аутентификации:', error);

      }

    }

  }

};

</script>

<style scoped>

.admin-login {

  max-width: 400px;

  margin: 0 auto;

}

h2 {

  text-align: center;

  margin-bottom: 20px;

}

form {

  margin-top: 20px;

}

input {

  width: 100%;

  padding: 8px;

  margin-bottom: 10px;

}

button {

  background-color: #007bff;

  color: #fff;

  border: none;

  padding: 10px 20px;

  border-radius: 5px;

  cursor: pointer;

}

button:hover {

  background-color: #0056b3;

}

</style>

# Компонент для страницы администратора (AdminPanel.vue):

<template>

  <div class="admin-panel">

    <h2>Панель администратора</h2>

    <div v-if="bookings.length">

      <div v-for="booking in bookings" :key="booking.id" class="booking">

        <p>ФИО: {{ booking.user\_full\_name }}</p>

        <p>Телефон: {{ booking.user\_phone }}</p>

        <p>Мастер: {{ booking.master\_name }}</p>

        <p>Дата и время: {{ booking.booking\_datetime }}</p>

        <p>Статус: {{ booking.status\_name }}</p>

        <select v-model="booking.id\_status" @change="updateStatus(booking)">

          <option v-for="status in statuses" :key="status.id" :value="status.id">{{ status.name }}</option>

        </select>

      </div>

    </div>

    <p v-else>Заявки отсутствуют.</p>

  </div>

</template>

<script>

import axios from 'axios';

export default {

  data() {

    return {

      bookings: [],

      statuses: [],

    };

  },

  async created() {

    try {

      const response = await axios.get('http://localhost:8000/admin/bookings', { withCredentials: true });

      if (response.status === 200) {

        this.bookings = response.data;

        this.getStatuses();

      } else {

        this.$router.push('/admin-login');

      }

    } catch (error) {

      console.error('Ошибка при загрузке данных:', error);

    }

  },

  methods: {

    async getStatuses() {

      try {

        const response = await axios.get('http://localhost:8000/statuses', { withCredentials: true });

        this.statuses = response.data;

      } catch (error) {

        console.error('Ошибка при загрузке статусов:', error);

      }

    },

    async updateStatus(booking) {

      try {

        const response = await axios.put(`http://localhost:8000/admin/bookings/${booking.id}`, {

          status\_id: booking.id\_status

        }, { withCredentials: true });

        if (response.status === 200) {

          alert('Статус обновлен');

        } else {

          alert('Ошибка при обновлении статуса');

        }

      } catch (error) {

        alert('Ошибка при обновлении статуса');

      }

    },

  },

};

</script>

<style scoped>

.admin-panel {

  max-width: 800px;

  margin: 0 auto;

}

.booking {

  border: 1px solid #ccc;

  padding: 15px;

  margin-bottom: 15px;

  border-radius: 5px;

}

h2 {

  text-align: center;

  margin-bottom: 20px;

}

p {

  margin: 5px 0;

}

select {

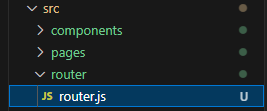
  margin-top: 10px;

}

</style>

После добавления всех компонентов, необходимо добавить маршрутизацию. Вот как это сделать:

В папке router (рисунок ниже) необходимо создать файл router.js и удалить index.js



В данный файл внести следующее

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

import Main from '@/pages/Main';

import Register  from '@/components/Register.vue'

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

import Bookings  from '@/components/Bookings.vue';

import NewBooking  from '@/components/NewBooking.vue';

import AdminPanel  from '@/components/AdminPanel.vue';

import AdminLogin  from '@/components/AdminLogin.vue';

const routes = [

  { path: '/', component: Main },

  { path: '/register', component: Register },

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

  { path: '/bookings', component: Bookings },

  { path: '/new-booking', component: NewBooking },

  { path: '/admin-login', component: AdminLogin },

  { path: '/admin-panel', component: AdminPanel },

]

const router = createRouter({

  history: createWebHistory(process.env.BASE\_URL),

  routes

})

export default router

Для проверки запустите приложение командой **npm run serve** и откройте в браузере страницу [**http://localhost:8080/**](http://localhost:8080/)

Теперь можно запустить сервер на nodejs (команда npm start) и клиент vue (команда npm run serve) и протестировать приложение.