Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«Национальный исследовательский университет ИТМО»
(Университет ИТМО)

Факультет прикладной информатики
Образовательная программа: Интеллектуальные системы в гуманитарной сфере
Направление подготовки: 45.03.03 Интеллектуальные системы в гуманитарной сфере

О Т Ч Е Т

по выполнению проектной работы по предмету “Фронтенд разработка”

Обучающиеся: Иржанова Юлия Ирназаровна, Мансурова Ульяна Дамировна

Санкт-Петербург,
2025

Тема: Разработка SPA на Vue 3 с маршрутизацией, внешним API (axios), компонентным подходом и composables.

Цель работы

Разработать одностраничное веб-приложение на Vue 3, подключить Vue Router, реализовать получение данных из внешнего API (axios), разделить интерфейс на компоненты и вынести повторяющуюся логику в composables.

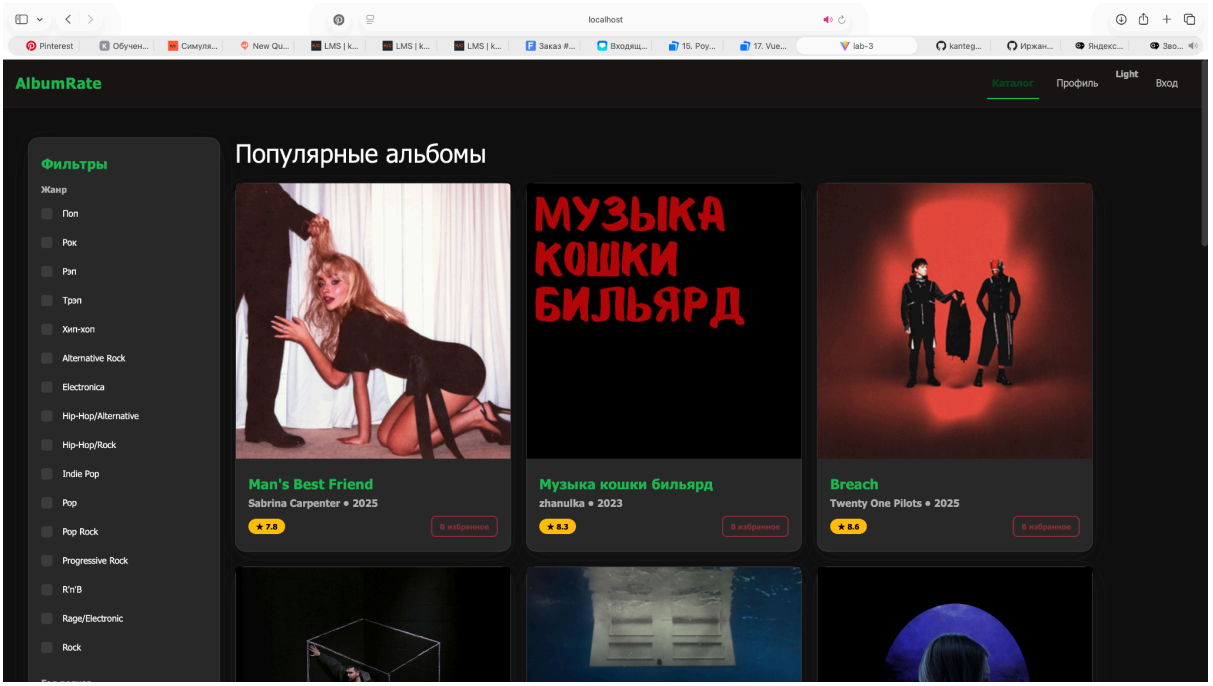
Используемые технологии

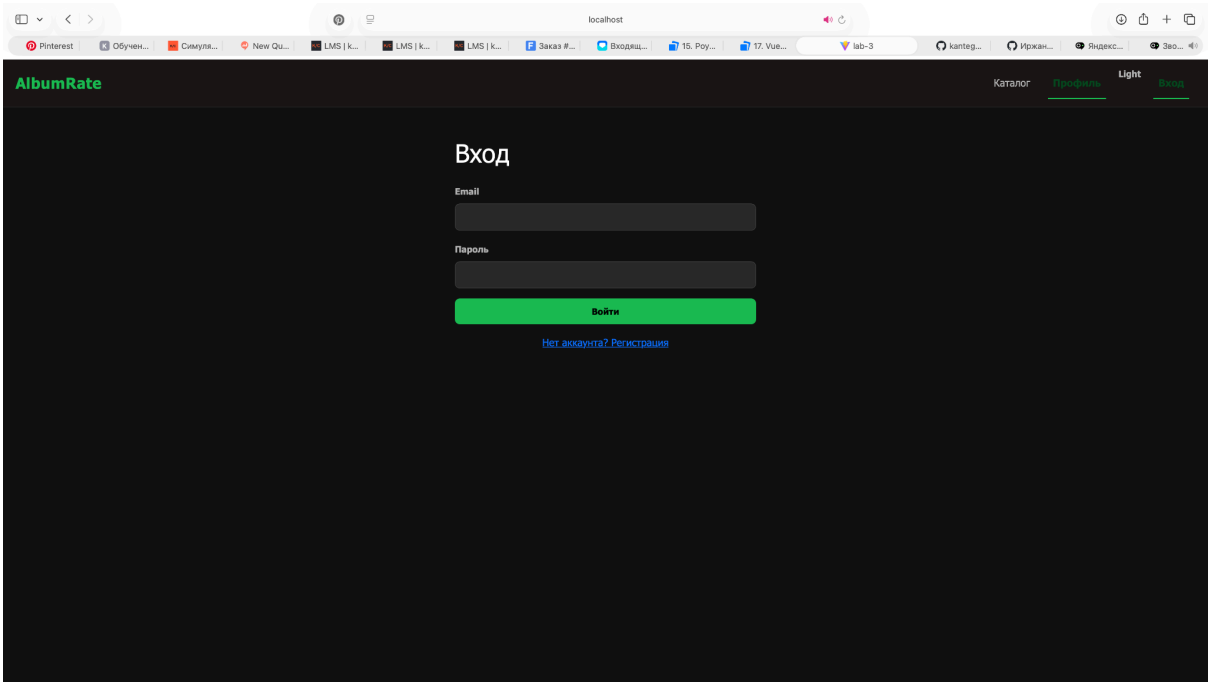
- Vue 3, Single File Components
- Vue Router (маршрутизация, navigation guards).
- Axios (HTTP-клиент), interceptors для токена.
- Bootstrap 5 (стили).

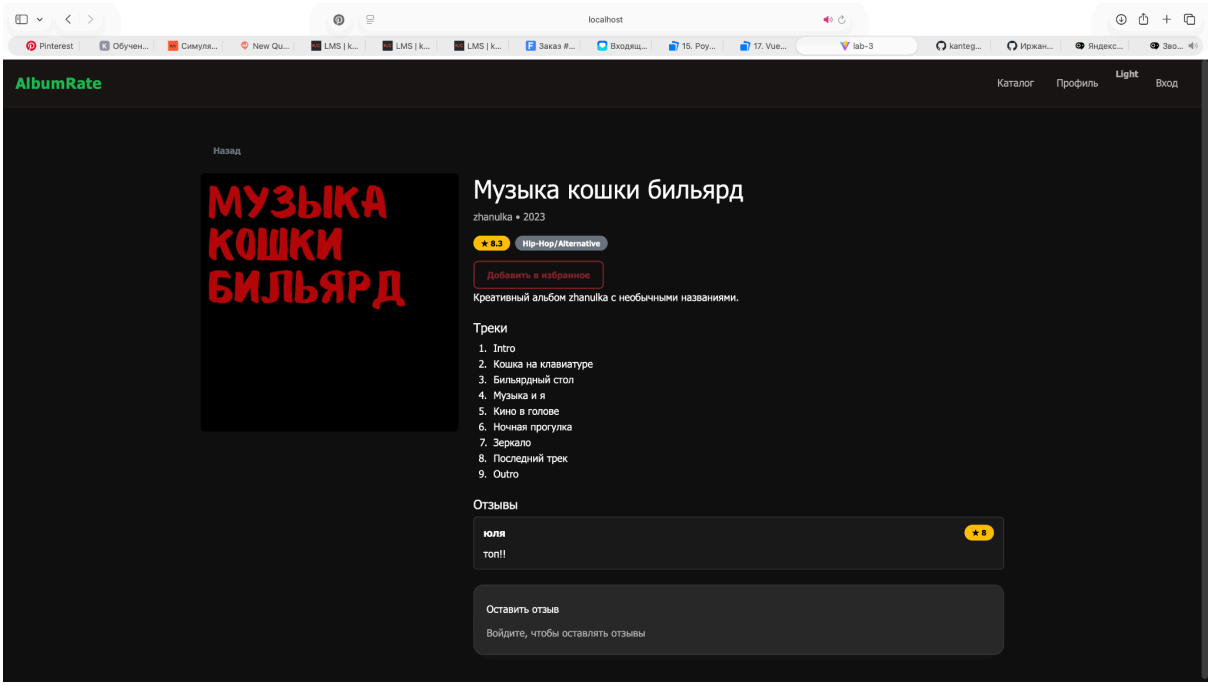
```
julirzhanova@Mac-mini-Julia lab-3 % npm run dev

> lab-3@0.0.0 dev
> vite

Port 5173 is in use, trying another one...

  VITE v7.3.1  ready in 127 ms

  ➜  Local:   http://localhost:5174/
  ➜  Network: use --host to expose
  ➜  press h + enter to show help
```

Приложение “AlbumRate” позволяет просматривать каталог музыкальных альбомов, открывать страницу альбома, добавлять альбомы в избранное, оставлять отзывы, а также работать с авторизацией и профилем пользователя.

Проект реализован как SPA: переходы между страницами выполняются через клиентский роутер без полной перезагрузки страницы.


localhost
AlbumRate
Каталог
Профиль
Light
Вход
Фильтры
Жанр
Поп
Рок
Рэп
Трэп
Хип-хоп
Alternative Rock
Electronica
Hip-Hop/Alternative
Hip-Hop/Rock
Indie Pop
Pop
Pop Rock
Progressive Rock
R'n'B
Rage/Electronic
Rock
Популярные альбомы
МУЗЫКА КОШКИ БИЛЬЯРД
Man's Best Friend
Sabrina Carpenter • 2025
★ 7.8
В избранное
Музыка кошки бильярд
zhanulka • 2023
★ 8.3
В избранное
Breach
Twenty One Pilots • 2025
★ 8.6
В избранное


localhost
AlbumRate
Каталог
Профиль
Light
Вход
Вход
Email
Пароль
Войти
Нет аккаунта? Регистрация



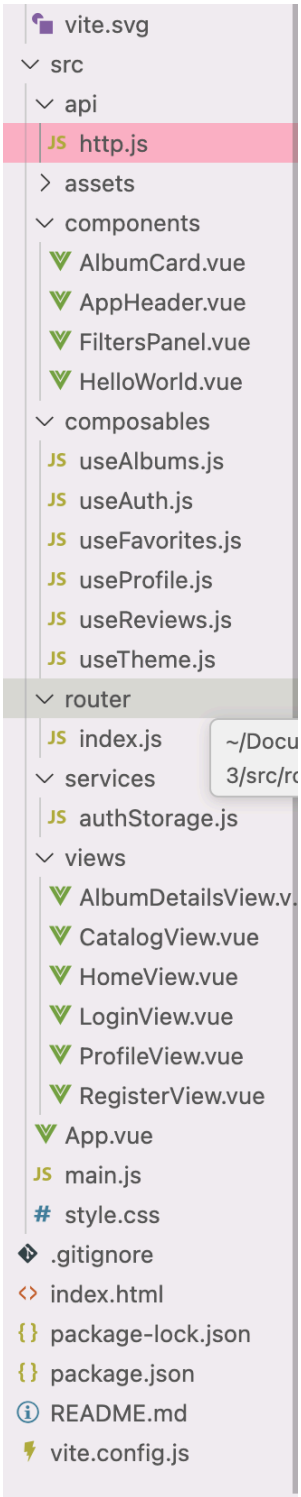
Структура проекта

Проект организован по стандартной структуре Vue-приложения: views (страницы), components (переиспользуемые UI-блоки), composables (логика/состояние), router (маршруты), api (axios-клиент), services (обёртки над хранилищем).

Фактическая структура (основные элементы):

- public/img — статические изображения обложек, доступны по URL /img/....
- src/main.js — точка входа: создание приложения и подключение роутера.
- src/App.vue — корневой layout, содержит AppHeader и <router-view>.
- src/router/index.js — маршруты и защита (guards).
- src/api/http.js — axios instance + interceptors.
- src/services/authStorage.js — работа с localStorage для токена/пользователя.
- src/components/* — AlbumCard.vue, AppHeader.vue, FiltersPanel.vue.
- src/views/* — CatalogView.vue, AlbumDetailsView.vue, LoginView.vue, RegisterView.vue, ProfileView.vue, HomeView.vue.

- src/composables/* — useAlbums.js, useAuth.js, useFavorites.js, useProfile.js, useReviews.js, useTheme.js.



src > App.vue

```vue
<template>
  <AppHeader />
  <main class="container-fluid my-5">
    <RouterView />
  </main>
</template>

<script setup>
import AppHeader from "./components/AppHeader.vue";
import { RouterView } from "vue-router";
</script>
```

Реализация требований задания

1) Подключение роутера
Vue Router подключён в приложении и используется для навигации между страницами.

Маршруты описаны в src/router/index.js и связывают URL с компонентами-страницами из папки src/views.

src > router > index.js > router > routes

```js
import { createRouter, createWebHistory } from "vue-router";
import { isAuthenticated } from "../services/authStorage";

import HomeView from "../views/HomeView.vue";
import CatalogView from "../views/CatalogView.vue";
import LoginView from "../views/LoginView.vue";
import RegisterView from "../views/RegisterView.vue";
import ProfileView from "../views/ProfileView.vue";
import AlbumDetailsView from "../views/AlbumDetailsView.vue";

const router = createRouter({
  history: createWebHistory(),
  scrollBehavior(to, from, savedPosition) {
    if (savedPosition) return savedPosition;
    return { top: 0 };
  },
  routes: [
    { path: "/", name: "home", component: HomeView },
    { path: "/catalog", name: "catalog", component: CatalogView },
    { path: "/login", name: "login", component: LoginView },
    { path: "/register", name: "register", component: RegisterView },
    { path: "/profile", name: "profile", component: ProfileView },
    { path: "/albums/:id", name: "album", component: AlbumDetailsView },
  ],
});

router.beforeEach((to) => {
  if (to.meta.requiresAuth && !isAuthenticated()) {
    return { path: "/login", query: { redirect: to.fullPath } };
  }

  if (to.meta.guestOnly && isAuthenticated()) {
    return { path: "/catalog" };
  }
});

export default router;
```

2) Защита маршрутов (meta + beforeEach)
Для защищённых страниц используется meta.requiresAuth, а перед переходами выполняется глобальная проверка в router.beforeEach.

Если пользователь не авторизован, доступ к защищённому маршруту блокируется и выполняется перенаправление на страницу входа.

```
router.beforeEach((to) => {
  if (to.meta.requiresAuth && !isAuthenticated()) {
    return { path: "/login", query: { redirect: to.fullPath } };
  }

  if (to.meta.guestOnly && isAuthenticated()) {
    return { path: "/catalog" };
  }
});

export default router;
```

3) Работа с внешним API через axios
Для запросов к API используется axios: общий клиент создан в src/api/http.js как axios.create({ baseURL }).

Загрузка альбомов реализована в composable useAlbums.js через http.get('/albums') и http.get('/albums/:id'), а состояние загрузки/ошибок

хранится в loading/error.

src > api > JS http.js > ...

```
import axios from "axios";
import { getToken } from "../services/authStorage";

export const http = axios.create({
  baseURL: "http://localhost:3000",
  headers: {
    "Content-Type": "application/json",
  },
});

http.interceptors.request.use((config) => {
  const token = getToken();
  if (token) {
    config.headers.Authorization = `Bearer ${token}`;
  }
  return config;
});

```

src > composables > JS useAlbums.js > ...

```
import { ref } from "vue";
import { http } from "../api/http";

export function useAlbums() {
  const albums = ref([]);
  const album = ref(null);
  const loading = ref(false);
  const error = ref(null);

  const fetchAlbums = async () => {
    loading.value = true;
    error.value = null;

    try {
      const { data } = await http.get("/albums");
      albums.value = data;
    } catch (e) {
      error.value = e?.message ?? "Failed to fetch albums";
    } finally {
      loading.value = false;
    }
  };

  const fetchAlbumById = async (id) => {
    loading.value = true;
    error.value = null;

    try {
      const { data } = await http.get(`/albums/${id}`);
      album.value = data;
    } catch (e) {
      error.value = e?.message ?? "Failed to fetch album";
    } finally {
      loading.value = false;
    }
  };

  return { albums, album, loading, error, fetchAlbums, fetchAlbumById };
}

```

4) Bearer-токен через interceptor axios
После авторизации токен сохраняется (через authStorage.js), а axios interceptor автоматически добавляет заголовок Authorization: Bearer <token> ко всем запросам.

Это уменьшает дублирование кода и централизует логику авторизации на уровне HTTP-клиента.

```
)
L   http.interceptors.request.use((config) => {
2     const token = getToken();
3     if (token) {
4       config.headers.Authorization = `Bearer ${token}`;
5     }
6     return config;
7   });
8
```

5) Компонентный подход
Интерфейс разделён на переиспользуемые компоненты: карточка альбома (AlbumCard.vue), шапка (AppHeader.vue), панель фильтров (FiltersPanel.vue).

Страницы (views) собирают компоненты в единые экраны и подключают composables для данных и действий.

6) Использование composables
Повторяющаяся логика вынесена в src/composables, что соответствует паттерну Composition API: composables позволяют композиционно собирать сложную логику из небольших функций.

В проекте реализованы composables: useAlbums, useAuth, useFavorites, useProfile, useReviews, useTheme (каждый отвечает за свою функциональность и может использоваться в нескольких местах)

Заключение

В ходе работы разработано SPA на Vue 3 с подключённым роутером, защитой маршрутов, работой с внешним API через axios, компонентной структурой и выделением повторяющейся логики в composables.

Структура проекта разделяет ответственность между слоями (views/components/composables/api/router), что упрощает поддержку и расширение приложения.

Вывод

В исследовании анализируются языковые стратегии самопрезентации в анкетах сайта знакомств, выявляя существенные различия между гендерными и возрастными группами. Установлено, что мужчины чаще используют в анкетах слова, связанные с социальным статусом, физическими параметрами и конкретными действиями («работаю», «рост», «курю», «зал»), концентрируясь на перечислении характеристик и формулировке запроса на партнёрство («ищу девушку»). В то же время женщины чаще акцентируют эмоциональную и коммуникативную составляющую, употребляя слова, связанные с поиском общения, взаимностью и общими интересами («ищу», «общение», «друзей», «люблю читать»). Эти различия отражают характерные для гендерных ролей паттерны самопредъявления: мужской фокус на статусных атрибутах против женского фокуса на построении эмоциональной связи и совместном досуге.

Возрастная динамика также оказалась значимой: у пользователей младше 20 лет (особенно у девушек) в анкетах преобладает лексика, связанная с учебой, увлечениями и неформальным общением («учусь», «аниме», «рисую», «лс»). После 20 лет профиль анкет смещается в сторону более серьёзных и осознанных формулировок: появляются темы работы, жизненных целей и поиска долгосрочных отношений («работаю», «серьёзные отношения», «хочется», «жизни»). Интересно, что длина анкет у мужчин существенно увеличивается с возрастом (с ≈20 до ≈26 слов), что может свидетельствовать о переходе от лаконичной самопрезентации к более развёрнутому формулированию ожиданий. У женщин этот показатель остаётся стабильно высоким во всех возрастных группах, что подтверждает их большую вовлечённость в текстовое самовыражение в рамках изучаемой платформы.