

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ ОРЕНБУРГСКОЙ ОБЛАСТИ
ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ СРЕДНЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«ОРЕНБУРГСКИЙ КОЛЛЕДЖ ЭКОНОМИКИ И ИНФОРМАТИКИ»
(ГАПОУ СПО ОКЭИ)**

ОТЧЁТ

ОКЭИ 09.02.07. 9024 06 УП

Учебная практика по профилю специальности

«Разработка модулей программного обеспечения для компьютерных систем»
(вид практики)

ГАПОУ «Оренбургский колледж экономики и информатики»
(место прохождения практики)

Количество листов: 16

Дата готовности: 06.03.2024 г

Разработал: Кондауров В.М

Руководитель Малышев А.О.

Соответствие отчета с заданием на практику _____
(ФИО. Подпись руководителя от учебного заведения)

Оренбург 2024

Содержание

Введение	3
1 Техническое задание	4
2 Логика работы программы	5
3 Руководство системного программиста	8
4 Руководство оператора	11
Заключение	16

					ОКЭИ 09.02.07. 9024 06 УП						
Изм.	Лист	№ докум.		Подп.	дата						
Разраб.	Кондауров В.М.					«Разработка модулей программного обеспечения для компьютерных систем»			Лит.	Лист	Листов
Провер.	Лукьяненко О.В.										
Руковод.	Малышев А.О.				Отделение программирования, группа 4пк1						

Введение

Радио - это один из самых распространенных и доступных источников информации и развлечений, который позволяет людям получать новости, слушать музыку, обсуждать текущие события и просто отдохнуть. В наше время, с развитием технологий и мобильных устройств, радио перестало быть только средством вещания, а превратилось в инструмент для создания социальных сетей, обмена мнениями и проведения общения.

Современные приложения для воспроизведения радио предлагают широкий спектр функций, которые улучшают пользовательский опыт и делают прослушивание радио более удобным и увлекательным, выбрать избранные радиостанции и получать уведомления о новых выпусках, а также управлять воспроизведением прямо с экрана смартфона или планшета.

Создание приложения для воспроизведения радио - это не только возможность освоить навыки разработки программного обеспечения, но и понять особенности работы с аудиоданными, интеграцию с различными сервисами и устройствами, а также разработку интерфейсов пользовательского интерфейса, которые будут удобными и интуитивными для пользователя.

Важно отметить, что создание приложения для воспроизведения радио не только требует навыков программирования, но и понимания рыночных требований, технологий и тенденций развития медиа-индустрии.

Это позволяет студентам не только приобрести практические навыки, но и развить стратегическое мышление и аналитическое мышление. Это представляет собой ценный опыт для студентов, позволяющий им освоить навыки разработки программного обеспечения и применить их на практике в интересной и актуальной области медиа-технологий.

1 Техническое задание

Описание приложения:

PWA приложение на React под названием "Куся Эфир" для прослушивания онлайн-радио и отображения информации о текущем треке.

Функциональные требования:

- Воспроизведение радио из списка;
- Поиск радио по названию;
- Возможность регулировать громкость;
- Возможность добавления радиостанций в избранное.;
- Регистрация;
- Авторизация.

Технологии и инструменты:

- Использование React для создания пользовательского интерфейса;
- Использование сервисного воркера для кэширования аудиофайлов и обеспечения оффлайн-режима;
- Использование IndexedDB для хранения информации о радиостанциях и избранных треках;
- Использование Web Audio API для работы с аудиофайлами.

Дизайн и пользовательский интерфейс:

- Создание привлекательного и интуитивно понятного пользовательского интерфейса;
- Использование адаптивного дизайна для поддержки различных устройств и разрешений экрана.

Тестирование:

- Проведение функционального тестирования для проверки работоспособности приложения;
- Проведение тестирования на различных устройствах и браузерах для обеспечения совместимости.

Производство и развертывание:

- Создание сборки приложения с помощью инструментов сборки, таких как Vite;
- Развертывание приложения на хостинге с поддержкой PWA.

Входные данные:

- Список избранных радио;
- Запрос радио эфиров;
- Желаемый уровень громкости.

Выходные данные

- Актуальный список избранных радио;
- Изменённый уровень громкости.

2 Логика работы программы

Приложение для воспроизведения радио через интернет имеет определенную логику работы, которая обеспечивает корректное функционирование и удобство использования для пользователей. Вот основные шаги и принципы, определяющие логику работы такого приложения:

- Подключение к интернету: Первоначальным шагом является установление соединения с интернетом, что позволяет приложению получать доступ радио потокам.
- Выбор радиостанции: Пользователь может выбрать желаемую радиостанцию из списка доступных в приложении.
- Прослушивание радиопередач: После выбора станции, приложение начинает воспроизведение аудиопотока с выбранной радиостанции, обеспечивая пользователю возможность слушать радиопередачи в реальном времени (Рис. 1).
- Управление воспроизведением: Пользователю предоставляются функциональные элементы управления, такие как пауза (Рис. 2) и громкость (Рис. 3), для удобного контроля за процессом прослушивания.
- Добавление в избранное: после авторизации (Рис. 4), пользователь может добавлять и удалять из списка избранного радиостанции (Рис. 5).

Таким образом, логика работы приложения для воспроизведения радио через интернет строится на последовательности действий от подключения к интернету до управления воспроизведением и предоставления дополнительных функций для комфортного использования пользователем.

```
useEffect(() => {  
  if (playingId === null || !audioRef.current) return;  
  
  const radioItem = radioList.filter(r => r.id === playingId)[0]  
  
  setRadioName(radioItem.title)  
  
  audioRef.current.src = radioItem.audio  
  audioRef.current.load()  
  audioRef.current.play()  
  setIsPause(false)  
  
}, [playingId])  
  
useEffect(() => {  
  audioRef.current.volume = volume  
}, [volume])
```

Рисунок 1 – Логика воспроизведения радио

Изм.	Лист	№ докум.	Подп.	Дата

ОКЭИ 09.02.07. 9024 06 УП

Лист

5

```
function onClickPlayPauseBtn() {
  isPause ? audioRef.current.play() : audioRef.current.pause()
  setIsPause(!isPause);
}
```

Рисунок 2 – Логика управлением остановки и воспроизведением

```
useEffect(() => {
  audioRef.current.volume = volume
}, [volume])
```

Рисунок 3 – Логика управлением громкостью

```
function onClickAuth() {
  if (loginValue === '' && passwordValue === '') return

  fetch(`${apiOrigin}/users/login`, {
    method: 'post',
    headers: {
      'Content-Type': 'application/json'
    },
    body: JSON.stringify({
      email: loginValue.trim(),
      password: passwordValue.trim()
    })
  })
  .then(res => res.json())
  .then(json => {
    console.log(json)
    if (json.error !== null) {
      setNotice(json.error)
      return
    }

    setNotice('')
    setUsername(json.data.login)
    setFavorites(JSON.parse(json.data.favorites_list))
    close()

    localStorage.setItem('user_name', json.data.login)
    localStorage.setItem('favorites', json.data.favorites_list)
    localStorage.setItem('user_id', json.data.id)
  })
}
```

Рисунок 4 – Логика авторизации

Изм.	Лист	№ докум.	Подп.	Дата

ОКЭИ 09.02.07. 9024 06 УП

Лист

6

```

function updateFavoritesInDB(radioId) {
  if (userName === null) {
    onClickFavoritListBtn()
    return
  }

  const favor = new Set(favorites)
  favor.has(radioId) ? favor.delete(radioId) : favor.add(radioId)
  const newFavor = Array.from(favor)
  setFavorites(newFavor)
  localStorage.setItem('favorites', JSON.stringify(newFavor))

  fetch(`${apiOrigin}/users/setFavorites`, {
    method: 'post',
    headers: {
      'Content-Type': 'application/json'
    },
    body: JSON.stringify({
      favorites: JSON.stringify(newFavor),
      id: +localStorage.getItem('user_id')
    })
  })
}

```

Рисунок 5 – Логика редактирования списка избранного

Изм.	Лист	№ докум.	Подп.	Дата

ОКЭИ 09.02.07. 9024 06 УП

Лист

7

3 Руководство системного программиста

Для работы с сервером потребуется наличие node js и знание js, ведь сервер написан на javascript.

Для работы с базой данных потребуется либо подключиться к СУБД (Рис. 6), либо имея экспортированную базу данных, развернуть на своей СУБД, потребуется знание SQL, и хорошее ориентирование в таблицах и знаниях их связей.

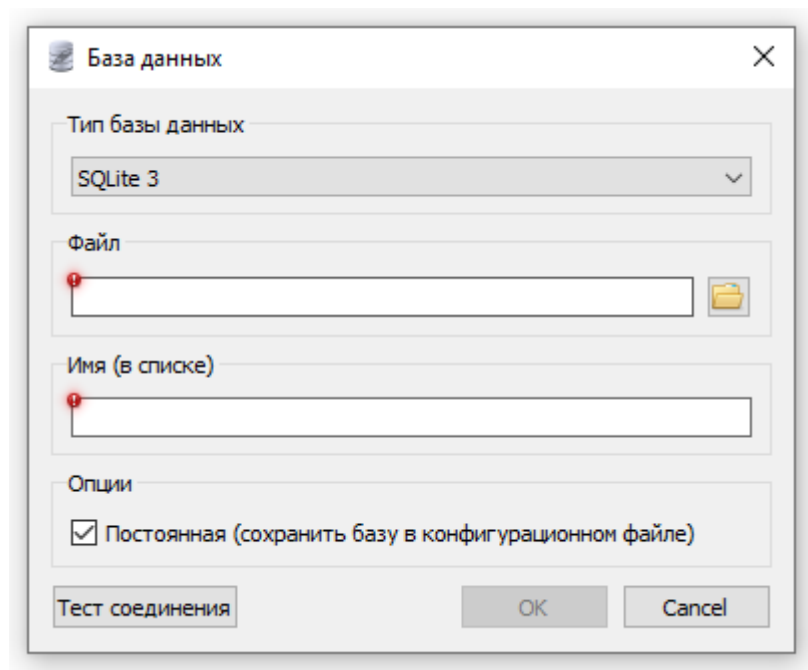


Рисунок 6 – Подключение к БД расположенной на сервере

Для редактирования клиентской части потребуется базовое знание html, css, и javascript, и логику взаимодействия клиента с сервером

Во frontend части веб приложения реализованы слушатели основных статических элементов.

Именованное классов реализовано по методологии БЭМ, грамотным и логичным образом, что позволяет интуитивно читать код и вносить правки в него.

При загрузке страницы реализована отправка fetch запроса на сервер для предварительного получения данных.

Для запуска к редактированию frontend части необходимо клонировать актуальную версию проекта из github

Затем нужно скачать пакеты для работы инструментов frontend разработки, это можно сделать, прописав команду «npm i» (Рис. 7).

Изм.	Лист	№ докум.	Подп.	Дата

ОКЭИ 09.02.07. 9024 06 УП

Лист

8


```

8      <link rel="stylesheet" href="css/style.css">
9      <script src="https://telegram.org/js/telegram-web-app.js"></script>
10     <script defer src="js/script.js"></script>
11   </head>
12
13   <body>
14     <section class="intro active" data-page="intro">...
27   </section>
28
29   <section class="write" data-page="write">
30     <div class="container">
31       <div class="breadcrumbs">
32         <div class="breadcrumbs__item"><span data-page-target=
33         <div class="breadcrumbs__item">Запись</div>
34       </div>

```

ПРОБЛЕМЫ ВЫХОДНЫЕ ДАННЫЕ КОНСОЛЬ ОТЛАДКИ ТЕРМИНАЛ ПОРТЫ

```

PS D:\PROJ\WEB\Сайты\okei_check> npm i
[#####.....] \ idealTree: timing idealTree Completed in 330ms

```

Рисунок 7 – Установка зависимых пакетов используя npm

После этой команды система установит все необходимые зависимости для разработки этого проекта, конфигурация зависимостей находится в файле package.json (Рис. 8)

```

package.json
1  {
2    "name": "kusya_stream",
3    "private": true,
4    "version": "0.0.0",
5    "type": "module",
6    "scripts": {
7      "dev": "vite",
8      "build": "vite build",
9      "lint": "eslint . --ext js,jsx --report-u
10     "preview": "vite preview"
11   },
12   "dependencies": {
13     "react": "^18.2.0",
14     "react-dom": "^18.2.0"
15   },
16   "devDependencies": {
17     "@types/react": "^18.2.55",
18     "@types/react-dom": "^18.2.19",
19     "@vitejs/plugin-react": "^4.2.1",
20     "eslint": "^8.56.0",
21     "eslint-plugin-react": "^7.33.2",
22     "eslint-plugin-react-hooks": "^4.6.0",
23     "eslint-plugin-react-refresh": "^0.4.5",
24     "sass": "^1.71.1",
25     "vite": "^5.1.0"
26   }
27 }

```

Рисунок 8 – Установка зависимых пакетов используя npm

После установки всех зависимостей можно запустить среду разработки комплексной frontend части, прописав команду «npm run dev» (Рис. 9).

```
PS D:\PROJ\React\kunya_stream> npm run dev

> kunya_stream@0.0.0 dev
> vite

VITE v5.1.3 ready in 2121 ms

→ Local:   http://localhost:5173/
→ Network: use --host to expose
→ press h + enter to show help

o
```

Рисунок 9 – Запуск инструментов разработки frontend части

После запуска будут собраны исходные файлы в папку public, и при разработке, будет постоянно отображаться актуальное состояние системы, а также автоматическое поточное обновление редактируемых элементов страница разработки

Для комфортной работы с клиентом, а точнее с получением данных БД, необходимо запустить сервер, это можно сделать с помощью команды «npm run server», после этого вы увидите по какому адресу работает сервер (Рис. 10).

```
PS D:\PROJ\Backend\kunya_stream_server> npm run dev

> kunya_stream_server@1.0.0 dev
> nodemon index.js

[nodemon] 3.1.0
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting `node index.js`
server started on 3000 port
```

Рисунок 10 – Запуск сервера node js

4 Руководство оператора

Для авторизации в приложении, нужно нажать на кнопку меню (Рис. 11).

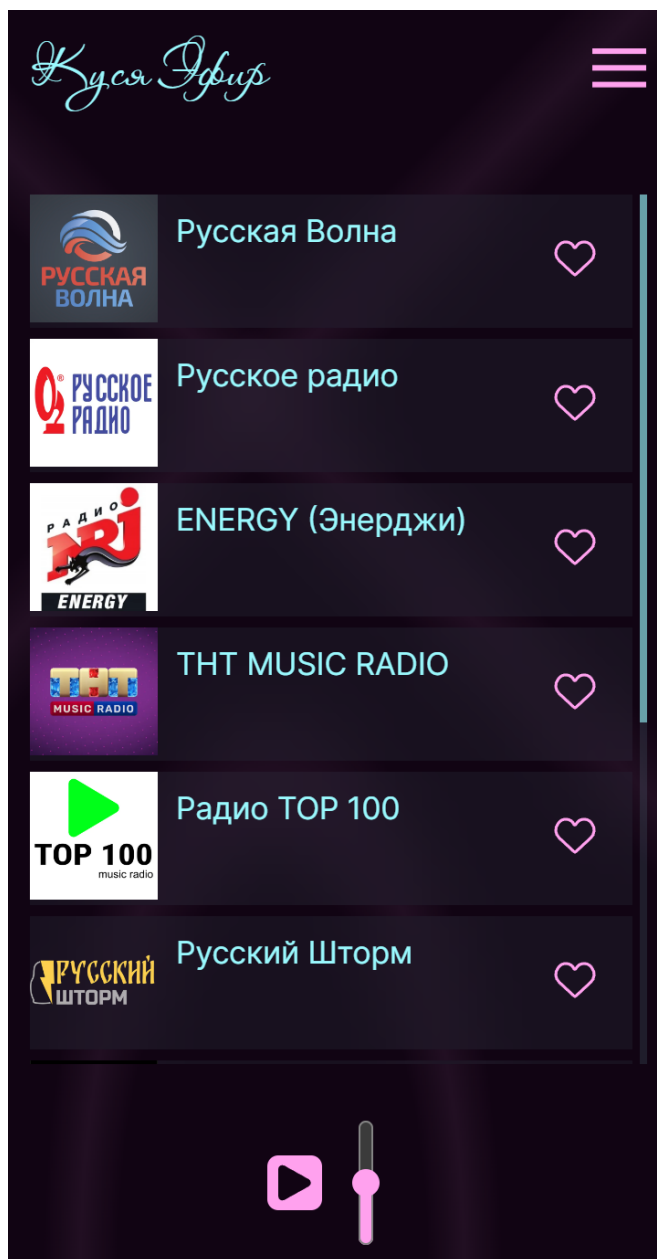


Рисунок 11 – Кнопка меню в верхнем углу

После этого нужно нажать на кнопку войти (Рис. 12).

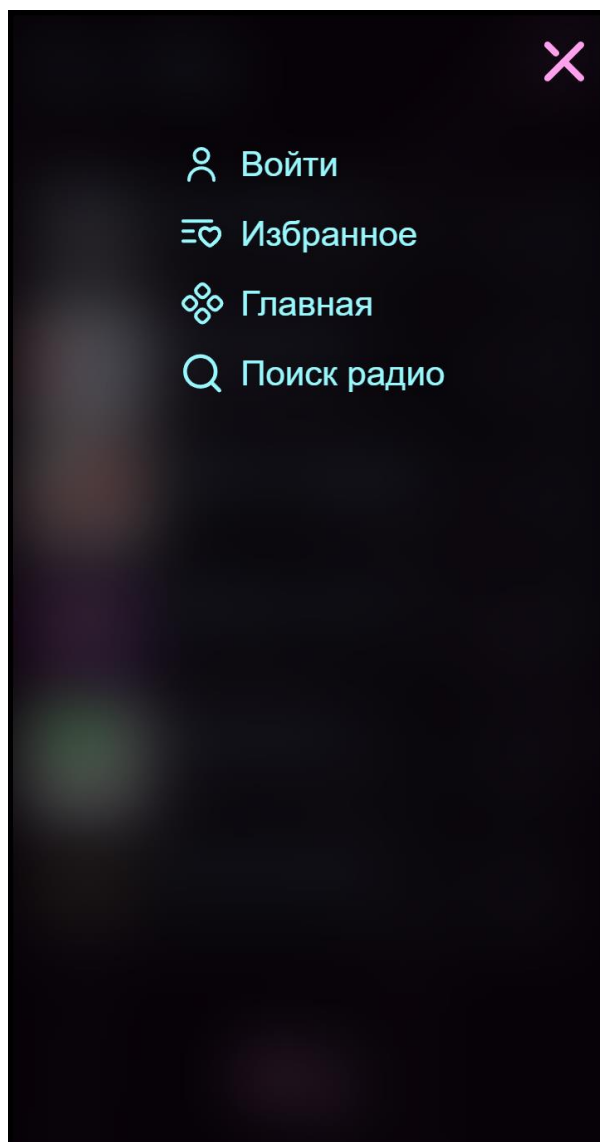


Рисунок 12 – Кнопка войти сверху списка меню

Затем, ввести почту и пароль в открывшемся окне и нажать на кнопку «Войти» (Рис. 13).

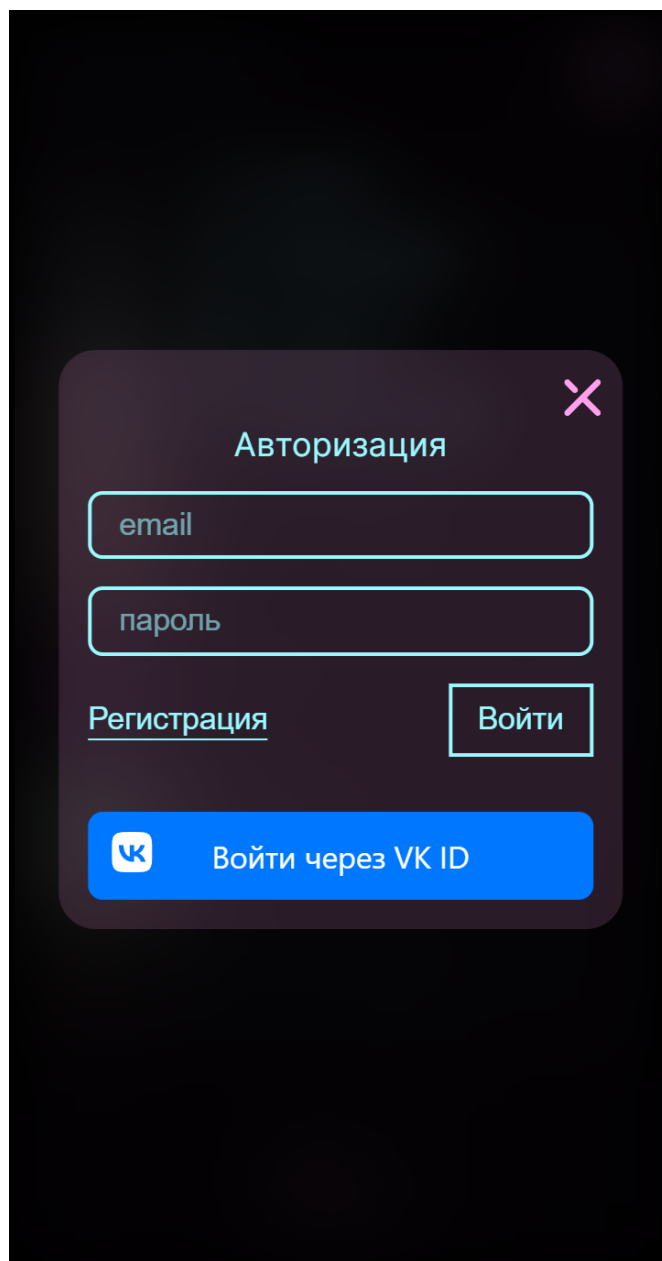


Рисунок 13 – Окно для авторизации

После этих действий вместо кнопки «Войти» будет строка с логином (Рис. 14).

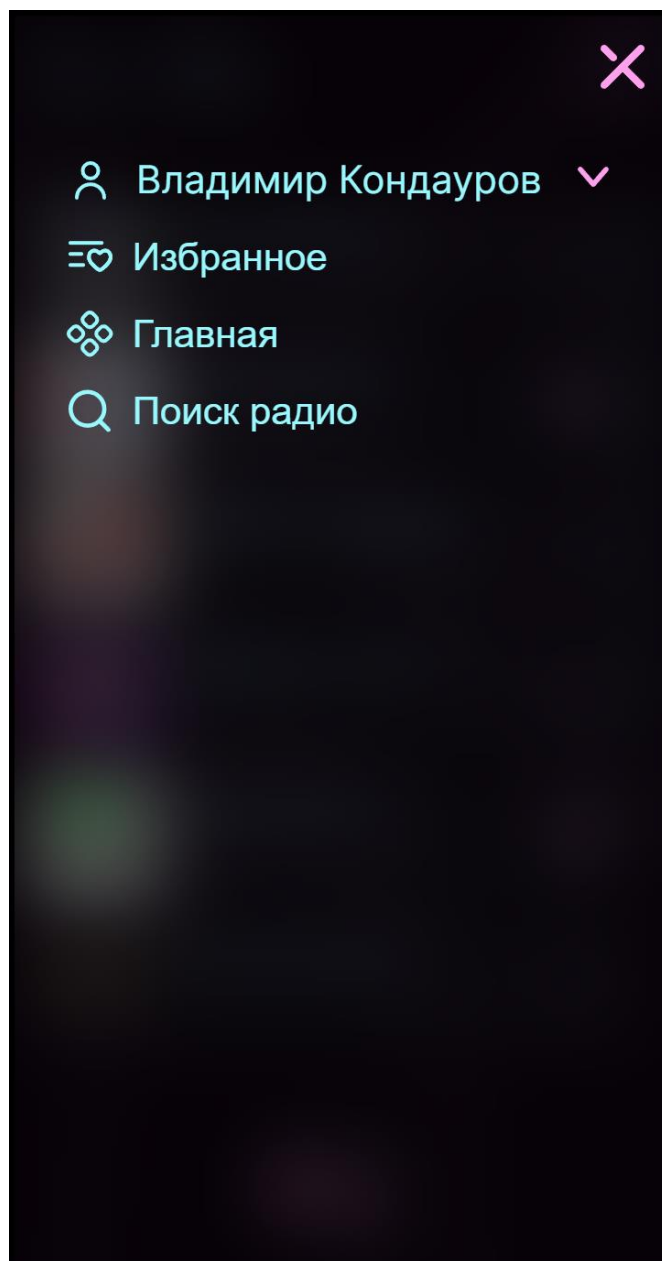


Рисунок 14 – Меню авторизованного пользователя

После авторизации можно добавлять радио в избранное, нажав на соответствующую кнопку (Рис. 15).

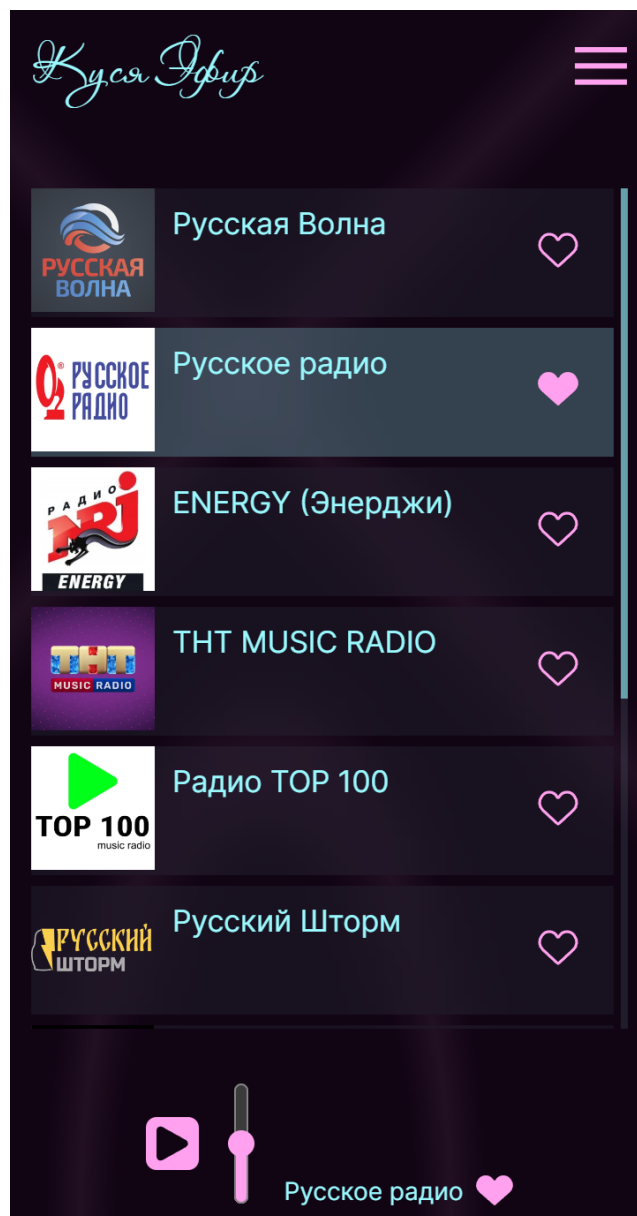


Рисунок 15 – Добавление радио в избранное

Остановка и воспроизведение радио, вместе с ползунком для изменения громкости находится в нижней части приложения (Рис. 15).

Изм.	Лист	№ докум.	Подп.	Дата

ОКЭИ 09.02.07. 9024 06 УП

Лист

15

Заключение

Разработка приложения для воспроизведения радио через интернет является сложным и многосторонним проектом, который требует знаний и навыков в различных областях, таких как разработка программного обеспечения, интеграция с различными сервисами и устройствами, а также понимание рыночных требований и тенденций развития медиа-индустрии.

В ходе учебной практики, мы провели анализ рыночных требований, выбрали технологии и инструменты для разработки приложения, интегрировали его с различными сервисами и устройствами, обеспечили безопасность и конфиденциальность данных пользователей, провели тестирование и отладку, а также создали руководства для системного программиста и оператора.

В результате, мы получили функциональное приложение для воспроизведения радио через интернет, которое может быть использовано для прослушивания радиопередач, что обеспечивает пользователям удобство и увлекательность в процессе воспроизведения радио.

Важно отметить, что создание приложения для воспроизведения радио через интернет - это не только возможность освоить навыки разработки программного обеспечения, но и понять особенности работы с аудиоданными, интеграцию с различными сервисами и устройствами, а также разработку интерфейсов пользовательского интерфейса, которые будут удобными и интуитивными для пользователя.

Важно также отметить, что создание приложения для воспроизведения радио через интернет не только требует навыков программирования, но и понимания рыночных требований, технологий и тенденций развития медиа-индустрии. Поэтому, учебная практика позволяет студентам не только приобрести практические навыки, но и развить стратегическое мышление и аналитическое мышление.