

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ  
ФЕДЕРАЦИИ  
Федеральное государственное бюджетное образовательное  
учреждение  
высшего образования  
**«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**  
**(ФГБОУ ВО «КубГУ»)**

**Факультет компьютерных технологий и прикладной  
математики  
Кафедра информационных технологий**

**ЛАБОРАТОРНАЯ РАБОТА №7**

Работу выполнил \_\_\_\_\_ А. А. Иванов  
(подпись)

Направление подготовки 02.03.03 Математическое обеспечение и  
администрирование информационных систем

Направленность Технология проектирования программного  
обеспечения

Руководитель \_\_\_\_\_ Н.Ю. Добровольская  
(подпись)

Краснодар  
2024

Тема: разработка программного обеспечения ПС.

Цель: приобретение навыков проектирования и разработки программной системы при помощи современных инструментальных средств.

### Задание

1. В соответствии с требованиями технического задания, разработанного при выполнении лабораторной работы №3, провести обоснованный выбор средства разработки специального ПО. Разработать схему общесистемного ПО.
2. В соответствии с требованиями технического задания, разработанного при выполнении лабораторной работы №3, а также проектными решениями, разработанными при выполнении лабораторных работ №4 и №5, разработать специальное программное обеспечение ПС.
3. Выполнить описание разработанных компонентов приложения в виде таблицы. Типы компонентов указать согласно таблице. Имена компонентов-файлов привести с указанием расширения.
4. Построить структурную схему разработанного приложения в виде диаграммы компонентов UML, выражающую взаимодействие его компонентов с компонентами БД в процессе функционирования приложения.
5. Запустить приложение на выполнение. Убедиться в соответствии результатов выполнения приложения требованиям, установленным в техническом задании. При обнаружении логических ошибок задокументировать их и устранить.

6. Представить экранные формы компонентов приложения, в том числе отчётов.
7. Проанализировать код приложения по критерию сложности.
8. Выполнить описание физических элементов ПС в виде таблицы. Типы элементов указать согласно таблице.
9. Построить диаграмму развёртывания UML, выражающую зависимости между узлами ПС и развёрнутыми на них компонентами из таблицы.

## 1 Выбор средств разработки специального ПО и схема общесистемного ПО

Выбор средств разработки был подробно изложен в лабораторной работе №3, так что перечислим выбранные средства разработки: IDE Visual Studio Code, IDE PyCharm, СУБД PostgreSQL (pgAdmin4), используемые ЯП – HTML, CSS, JavaScript, библиотека для JavaScript – React, фреймворк Django Rest Api Framework.

Структурная схема разработанного ПО представлена на рисунке 1.

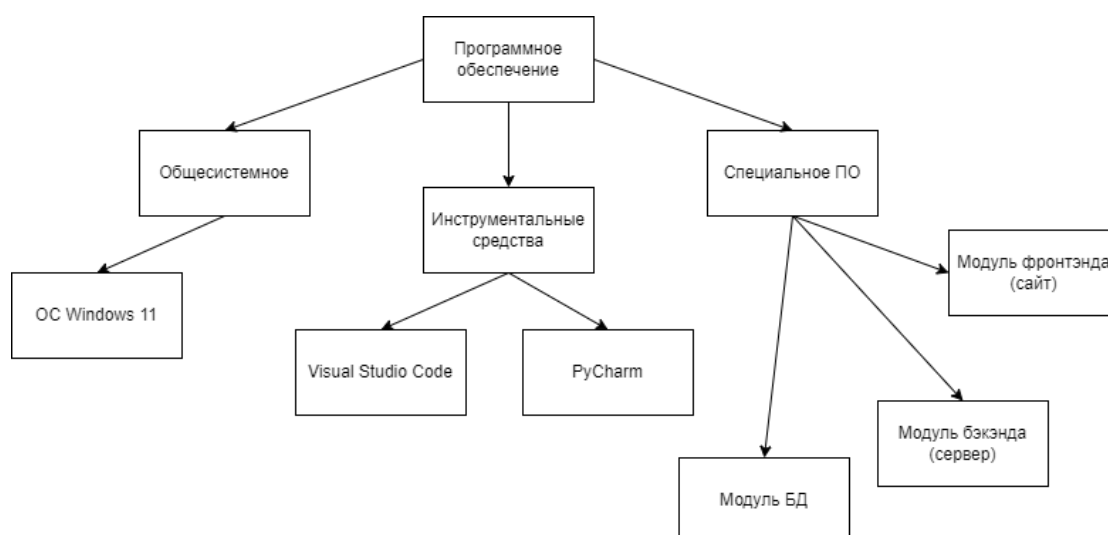


Рисунок 1 – структурная схема ПО

## 2 Разработанное специальное программное обеспечение ПС

Разработанное специальное программное обеспечение является веб-сервисом: объект информатизации – образовательный центр "ТехноКод". Информатизируемый бизнес-процесс - "Запись на курсы". На рисунках 2-10 представлен пользовательский интерфейс разработанного ПС, а также фрагмент базы данных.

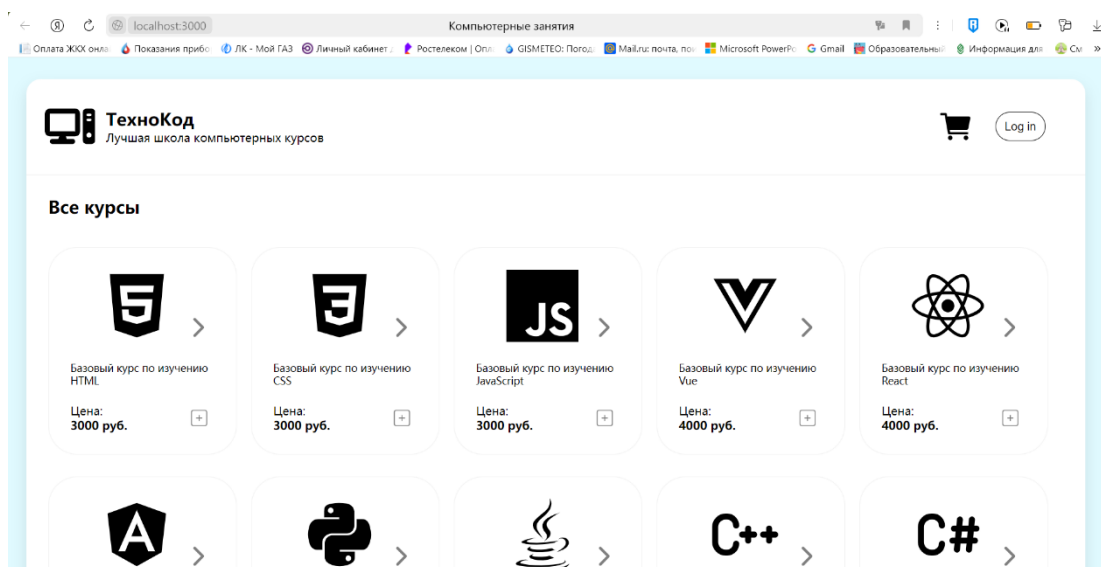


Рисунок 2 – вид главной страницы сайта

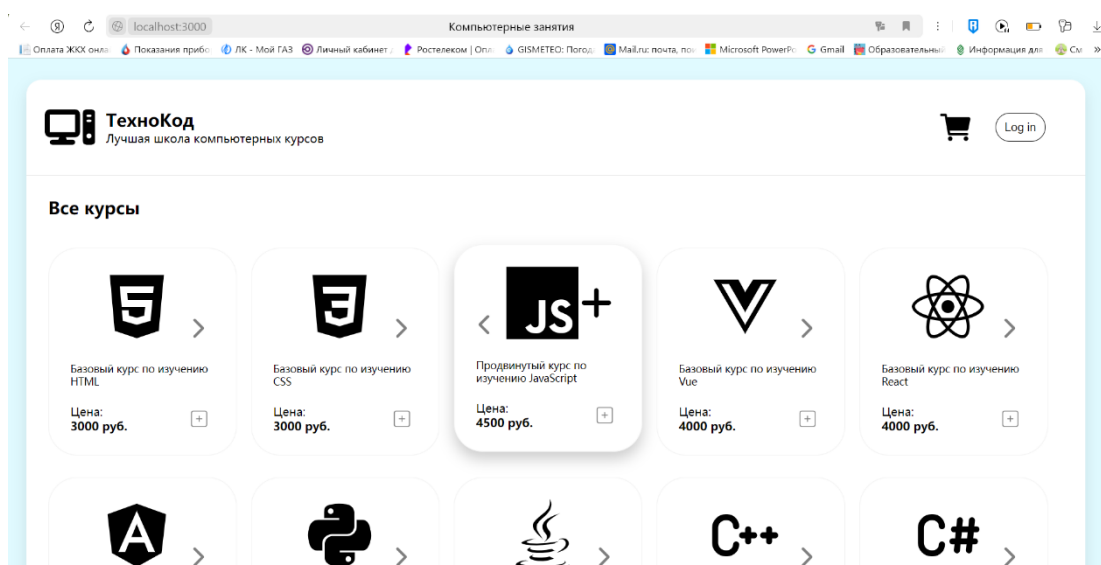


Рисунок 3 – вид главной страницы сайта (изменённая цена курса)

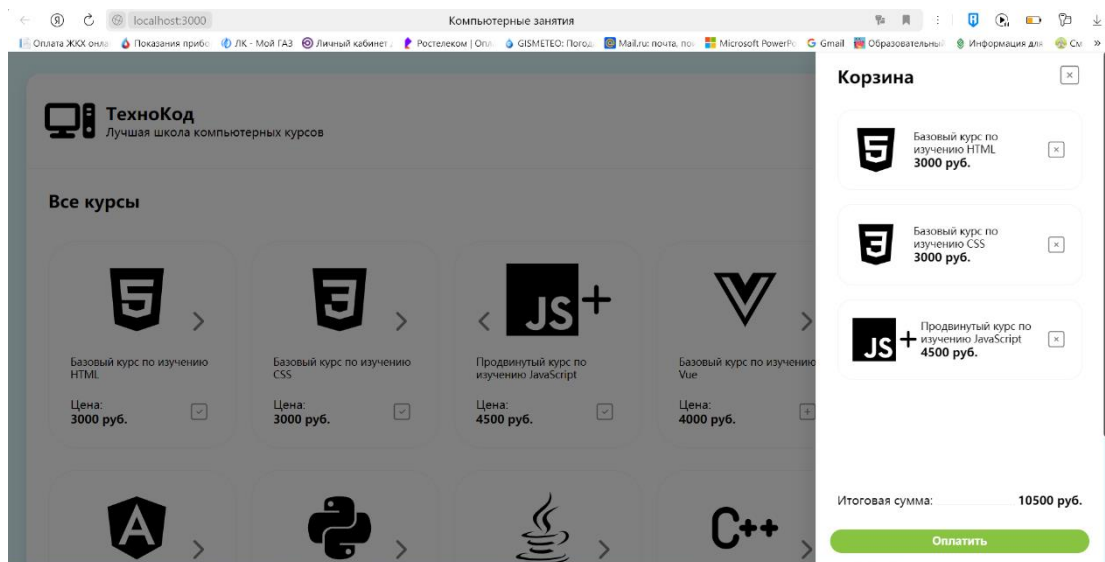


Рисунок 4 – вид корзины сайта

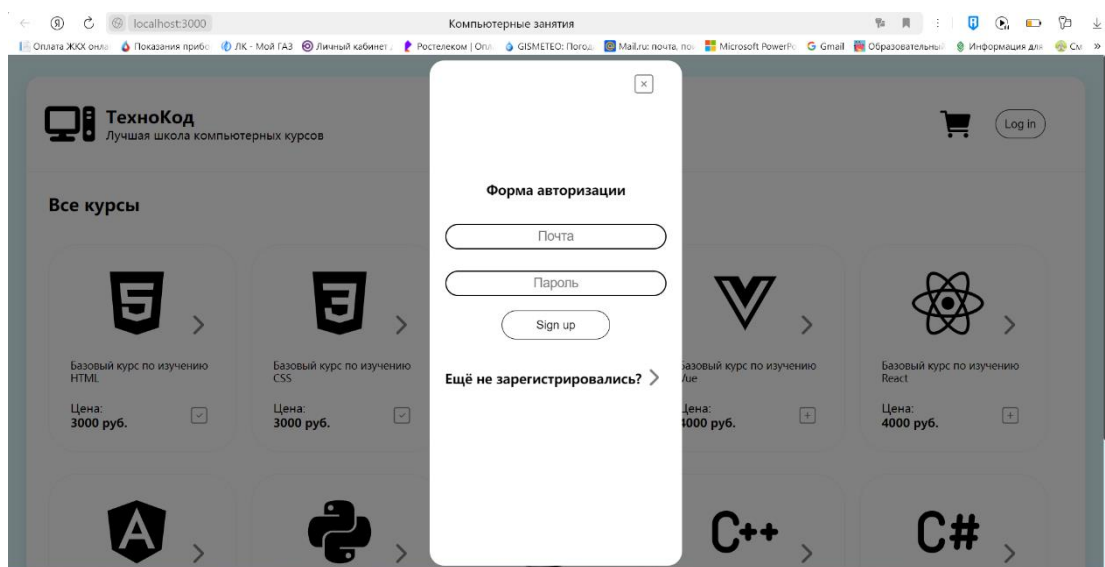


Рисунок 5 – вид формы авторизации

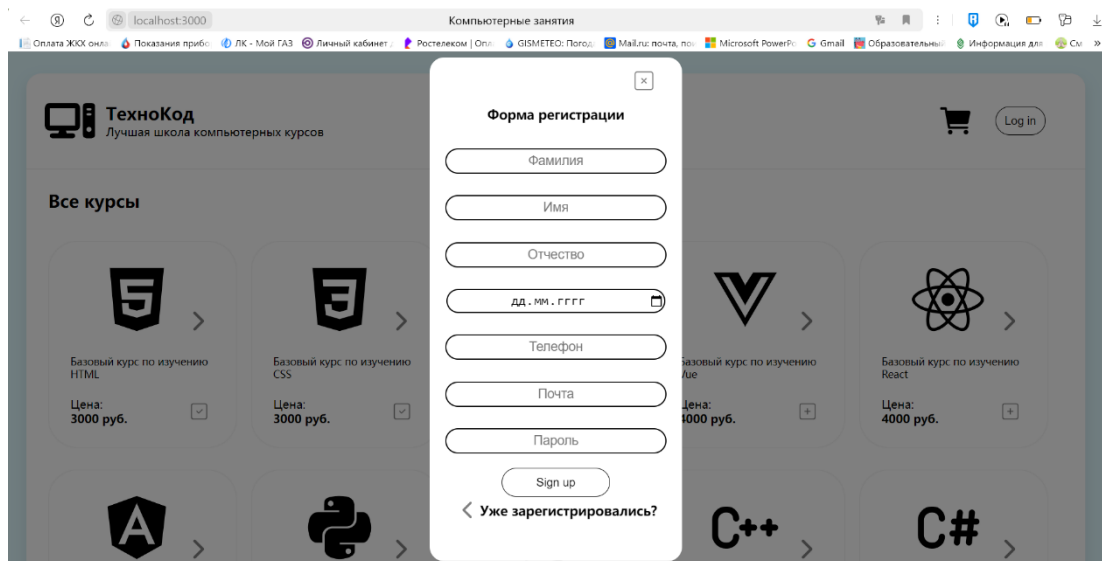


Рисунок 6 – вид формы регистрации

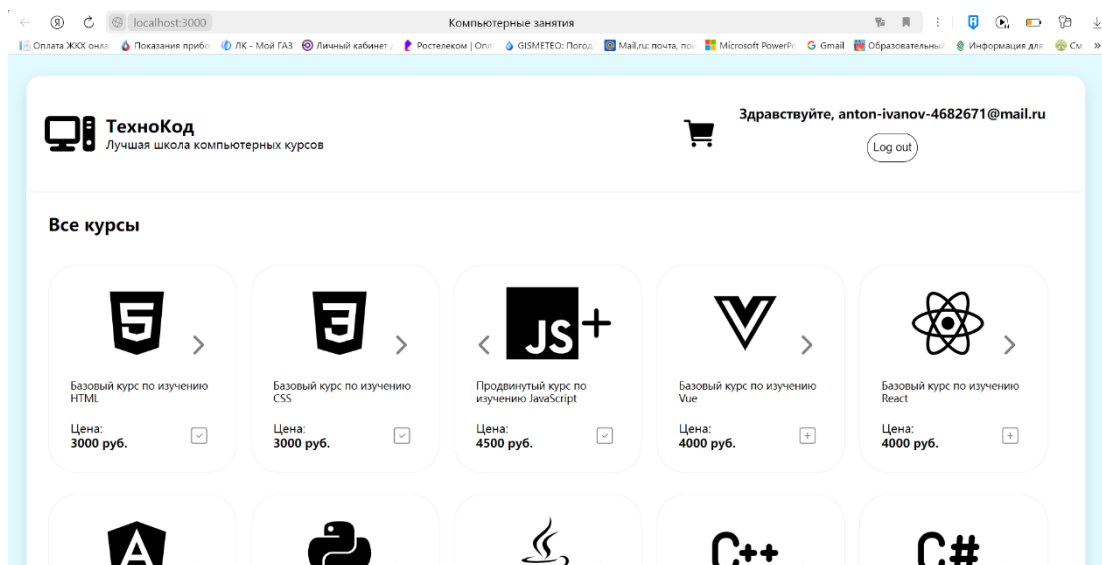


Рисунок 7 – вид главной страницы сайта после входа в систему

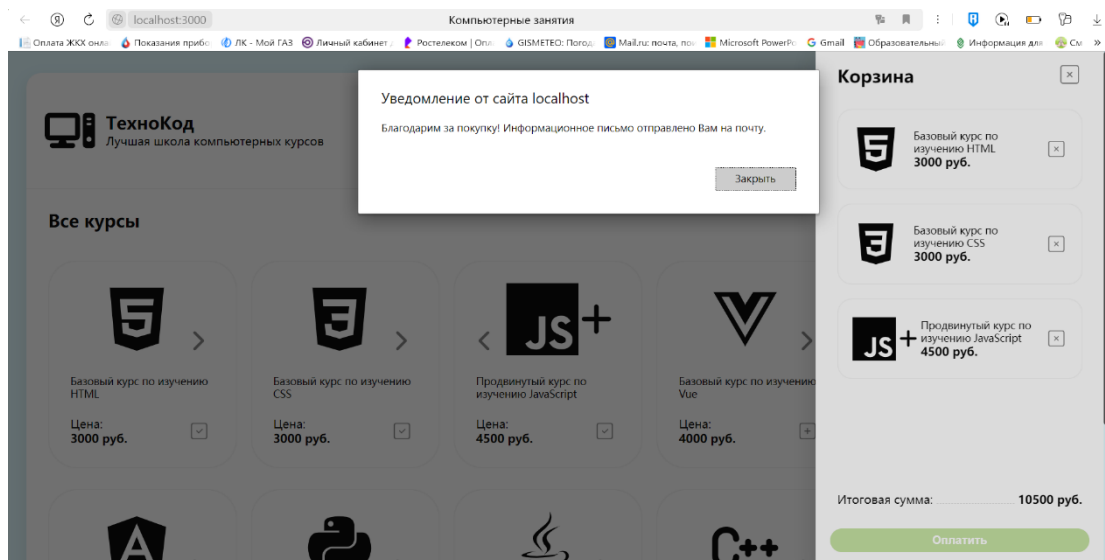


Рисунок 8 – вид уведомления после покупки

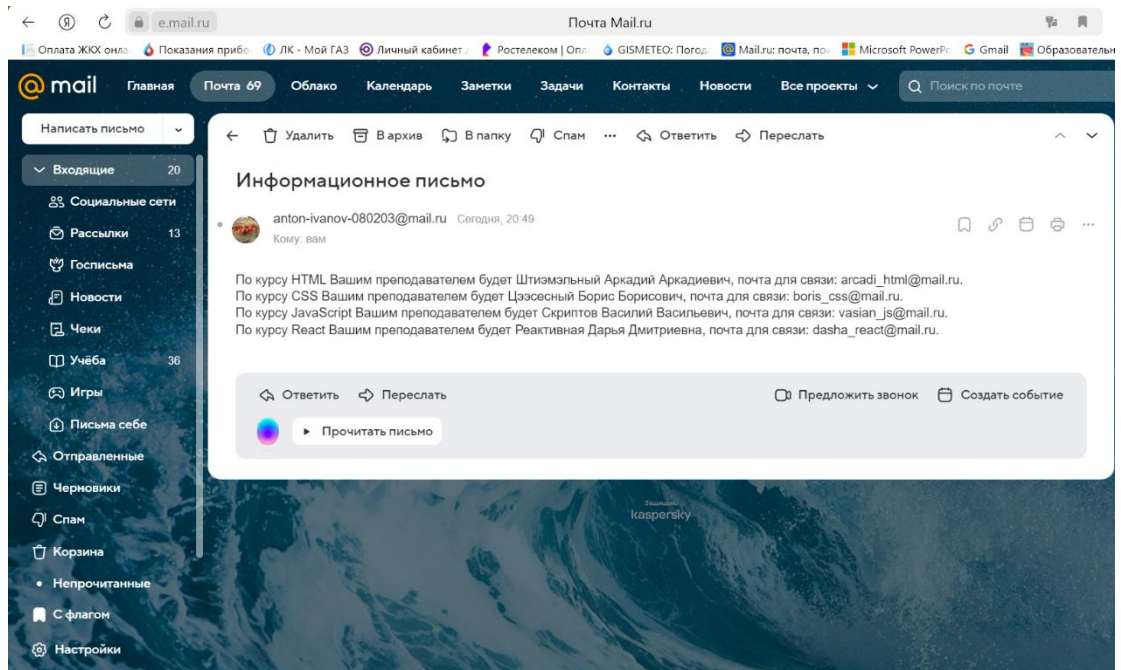


Рисунок 9 – вид информационного сообщения



Query: `SELECT * FROM public.listeners ORDER BY id ASC`

id	listeners_surname	listeners_name	listeners_patronymic	listeners_date_of_birth	listeners_telephone	listeners_e_mail	listeners_password
1	Иванова	Антон	Александрович	2003-02-08	89181214590	anton-ivanov-4582671@mail.ru	b08a640bc885300ef5229428ec6ca4da53b1

Рисунок 10 – вид таблицы пользователей

Query: `SELECT * FROM public.journal_of_academic_performance ORDER BY id ASC`

id	journal_of_academic	journal_of_academic	journal_of_academic	journal_of_academic	journal_of_academic	journal_of_academic	journal_of_academic	journal_of_academic_performance_teachers_id
1	3000	0	0	0	0	0	1	1
2	3000	0	0	0	0	0	1	2
3	3000	0	0	0	0	0	1	3
4	6000	0	0	0	0	0	1	5

Рисунок 11 – вид таблицы журнала успеваемости

### 3 Описание разработанных компонентов приложения

В таблице 1 перечислены компоненты приложения (файлы).

№	Имя	Тип	Описание
1	index.js	source	Базовый файл-скрипт, который отвечает за запуск других компонентов-скриптов
2	index.css	source	Базовый файл стилей, отвечает за стилизацию объектов в App.js
3	App.js	source	Главный файл-скрипт, который управляет рендерингом прочих компонентов
4	Lesson.js	source	Компонент, который отвечает за реализацию курсов как карточек на сайте
5	Lesson.module.css	source	Файл стилей, отвечает за стилизацию объектов в Lesson.js
6	Header.js	source	Компонент, который отвечает за реализацию шапки сайта (панель управления)
7	Header.module.css	source	Файл стилей, отвечает за стилизацию объектов в Header.js
8	Drawer.js	source	Компонент, который отвечает за реализацию корзины, в которой отображаются выбранные курсы
9	Drawer.module.css	source	Файл стилей, отвечает за стилизацию объектов в Drawer.js
10	Form.js	source	Компонент, который отвечает за реализацию форм авторизации и регистрации
11	Form.module.js	source	Файл стилей, отвечает за стилизацию

			объектов в Form.js
12	index.html	source	Базовый файл вёрстки, который отвечает за общее отображение страницы
13	settings.py	source	Главный файл настроек Django-сервера, отвечает за подключение к БД, настройку отправки сообщений, настройку cors-заголовков
14	models.py	source	Главный файл для работы с моделями таблиц базы данных
15	urls.py	source	Главный файл, который отвечает за переадресацию запросов (направляет запрос на корректную функцию-обработчик)
16	views.py	source	Главный файл, который реализует весь функционал сервера (содержит все функции для работы с запросами)
17	База данных	block	Блок, который отвечает за хранение всех данных

#### 4 Диаграмма компонентов

Связь компонентов представлена на рисунке 12.

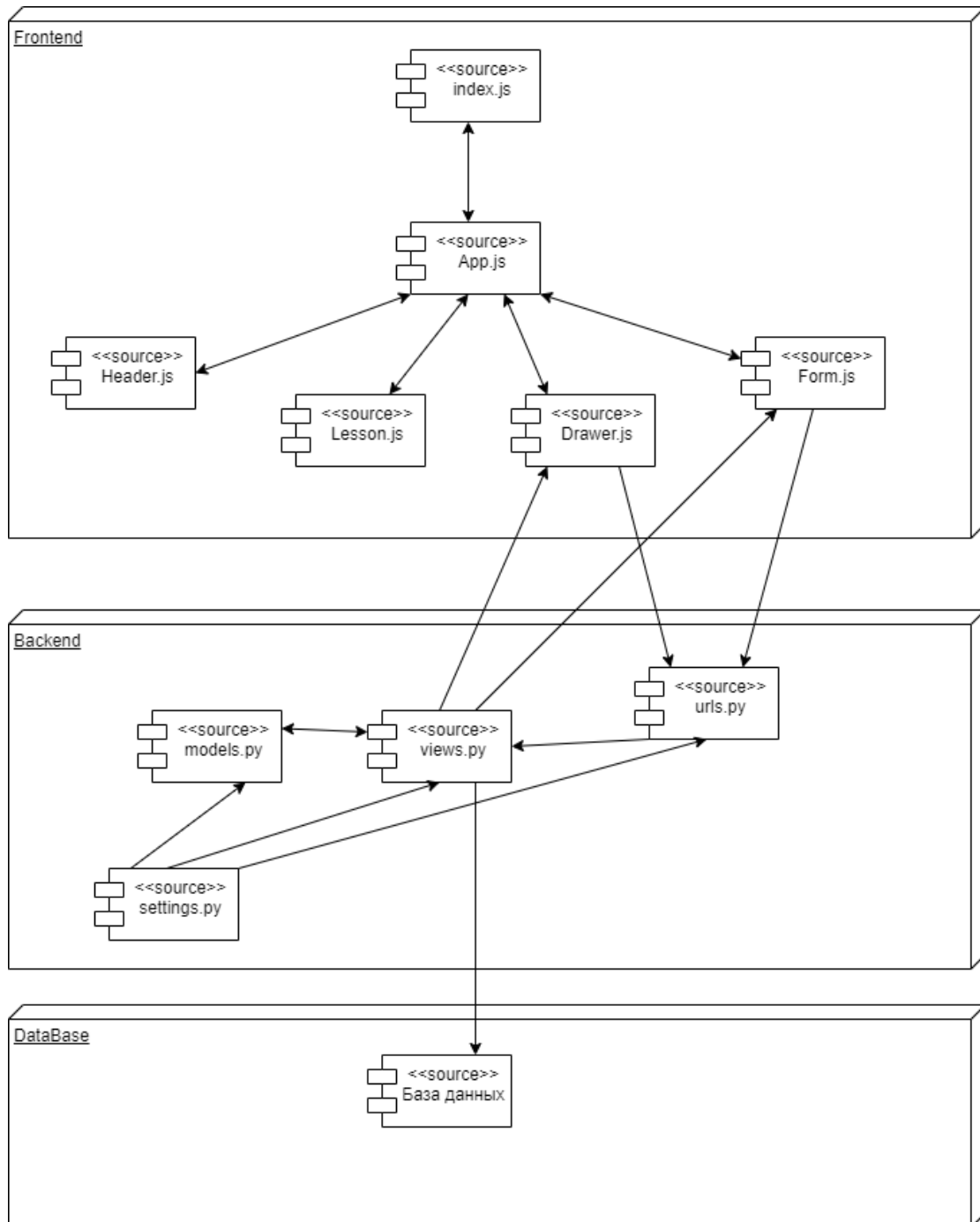


Рисунок 12 – связь компонентов

## **5 Запуск и проверка приложения**

После окончания разработки приложения было запущено и протестировано. Ошибки, который были обнаружены, были исправлены. На данном этапе разработки приложение выполняет свой заявленный спектр задач.

## **6 Сложность кода приложения**

Данное приложение представляет собой архитектуру средней сложности из-за выбранного стека технологий, однако подвержено улучшению и расширению.

В разработанном приложении:

- 16 основных файлов (в самом приложении несколько тысяч файлов, однако 16 указанных – те, в которых писался программный код, остальные же файлы представляют собой библиотеки и базовые файлы при создании React- и Django-приложений);
- 100 (примерно) объектов (те объекты и переменные, который были явно прописаны в вышеуказанных 16 файлах);
- 12 состояний рендеринга;
- 22 функции.

## 7 Диаграмма развёртывания

Диаграмма развёртывания разработанного ПО указана на рисунке 13.

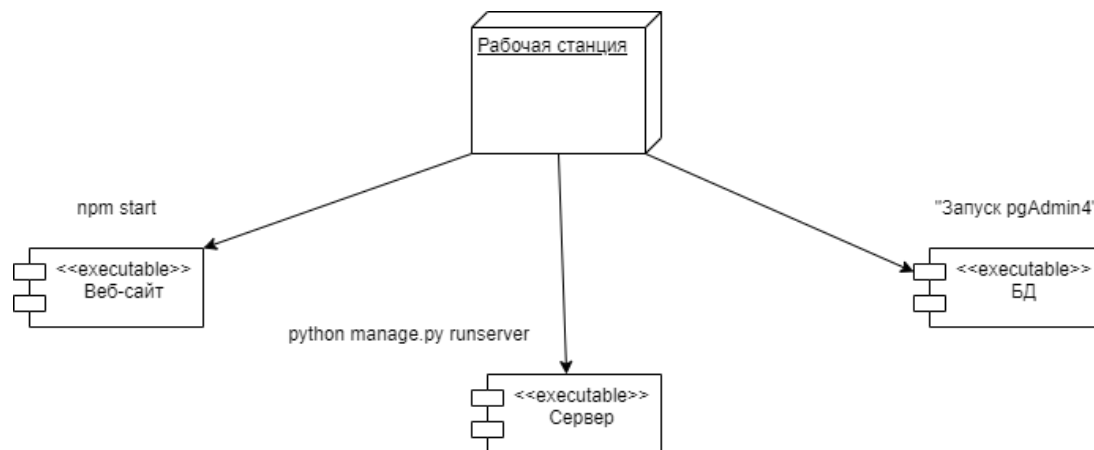


Рисунок 13 – диаграмма развёртывания ПО