

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение
высшего образования
«Курганский государственный университет»

Кафедра программного обеспечения автоматизированных систем

ТЕХНОЛОГИИ РАЗРАБОТКИ WEB-ПРИЛОЖЕНИЙ

Методические указания
к выполнению лабораторных работ и курсовых проектов
для студентов направлений 09.03.03, 09.03.04

Курган 2024

Кафедра: «Программное обеспечение автоматизированных систем».

Дисциплина: «Технологии разработки web-приложений».

Направления: 09.03.04 «Программная инженерия»,
09.03.03 «Прикладная информатика».

Составил: канд. техн. наук, доцент А. М. Семахин.

Печатается в соответствии с планом издания, утверждённым
методическим советом университета «25» декабря 2023 г.

Утверждены на заседании кафедры «15» ноября 2024 г.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	5
1 PHP-ФРЕЙМБОРК LARAVEL	5
1.1 Установка программного обеспечения	5
1.2 Создание и запуск пустого проекта веб-сайта	5
1.3 Создание контроллеров и операций	6
1.4 Создание веб-маршрутов	7
1.5 Создание миграций	7
1.6 Создание моделей	8
1.7 Шаблоны	8
1.8 Разграничение доступа	8
1.9 Лабораторная работа №1 «Веб-сайт коммерческих объявлений»	9
1.9.1 Варианты заданий	9
1.9.2 Методические указания	10
1.9.3 Контрольные вопросы	10
1.10 Лабораторная работа №2 «Веб-сайт фирмы, осуществляющей продажу товаров»	11
1.10.1 Варианты заданий	11
1.10.2 Методические указания	12
1.10.3 Контрольные вопросы	12
2 ФРЕЙМБОРК ANGULAR	13
2.1 Установка программного обеспечения	13
2.2 Создание и запуск пустого проекта веб-сайта	14
2.3 Компоненты в Angular	16
2.4 Декораторы в Angular	17
2.5 Сервисы в Angular	18
2.6 Маршрутизация в Angular	18
2.7 Связывание данных в Angular	19
2.8 Директивы в Angular	20
2.9 Лабораторная работа №3 «Веб-приложение “Выдающиеся деятели России”»	20
2.9.1 Варианты заданий	21
2.9.2 Методические указания	22
2.9.3 Контрольные вопросы	22
2.10 Лабораторная работа №4 «Веб-приложение “Список покупок”»	22
2.10.1 Варианты заданий	23

2.10.2 Методические указания	23
2.10.3 Контрольные вопросы	24
2.11 Лабораторная работа №5 «Веб-приложение “Трекер привычек”»	24
2.11.1 Варианты заданий	24
2.11.2 Методические указания	28
2.11.3 Контрольные вопросы	29
2.12 Лабораторная работа №6 «Веб-сайт электронной коммерции»	29
2.12.1 Варианты заданий	29
2.12.2 Методические указания	30
2.12.3 Контрольные вопросы	30
3 КУРСОВОЙ ПРОЕКТ	31
3.1 Назначение, цели и задачи курсового проекта	31
3.2 Требования к курсовому проекту	31
3.2.1 Требования к функциональным характеристикам	31
3.2.2 Требования к эксплуатационным характеристикам	32
3.2.3 Требования к программному обеспечению	32
3.3 Варианты заданий	32
3.3.1 Задание 1	32
3.3.2 Задание 2	33
3.3.3 Задание 3	33
3.4 Контрольные вопросы	34
ЗАКЛЮЧЕНИЕ	35
БИБЛИОГРАФИЧЕСКИЙ СПИСОК	35

ВВЕДЕНИЕ

Дисциплина «Технологии разработки web-приложений» имеет целью дать студентам теоретические знания и практические навыки в проектировании и реализации веб-сайтов.

Предмет дисциплины – технологии разработки веб-приложений.

Задачи дисциплины – дать представление о принципах веб-программирования и основных этапах разработки веб-приложений, сформировать компетенции и практические навыки.

Лабораторные работы включают 32 часа аудиторных занятий.

Методические указания содержат теоретическое обоснование, варианты заданий для выполнения лабораторных работ и курсовых проектов, контрольные вопросы по дисциплине «Технологии разработки web-приложений».

Методические указания разработаны в соответствии с требованиями государственного образовательного стандарта по подготовке бакалавров по направлениям 09.03.04 «Программная инженерия» и 09.03.03 «Прикладная информатика».

1 PHP-ФРЕЙМВОРК LARAVEL

1.1 Установка программного обеспечения

Установка программного обеспечения включает этапы:

1 Установка исполняющей среды PHP (<https://www.php.net>).

2 Установка *Composer* (<https://getcomposer.org>), в процессе которой проверить (указать) путь к файлу *php.exe*.

3 Установка утилиты *Laravel*, служащей для создания новых проектов. Открыть командную строку и набрать команду

Composer global require Laravel/installer [1].

1.2 Создание и запуск пустого проекта веб-сайта

Создание и запуск пустого проекта веб-сайта включает этапы:

1 Запустить командное окно.

2 Создать папку и перейти в нее.

3 Создать проект, используя команду *laravel new имя проекта*.

4 Перейти в папку проекта.

5 Установить библиотеку *laravel/ui*, используя команду *composer require Laravel/ui*.

6 Перейти в папку проекта.

7 Запустить отладочный сервер *PHP*, используя команду *php artisan serve*.

8 Запустить веб-обозреватель и перейти по интернет-адресу <http://localhost:8000>

Веб-обозреватель выведет страницу пустого веб-сайта (рисунок 1.1).

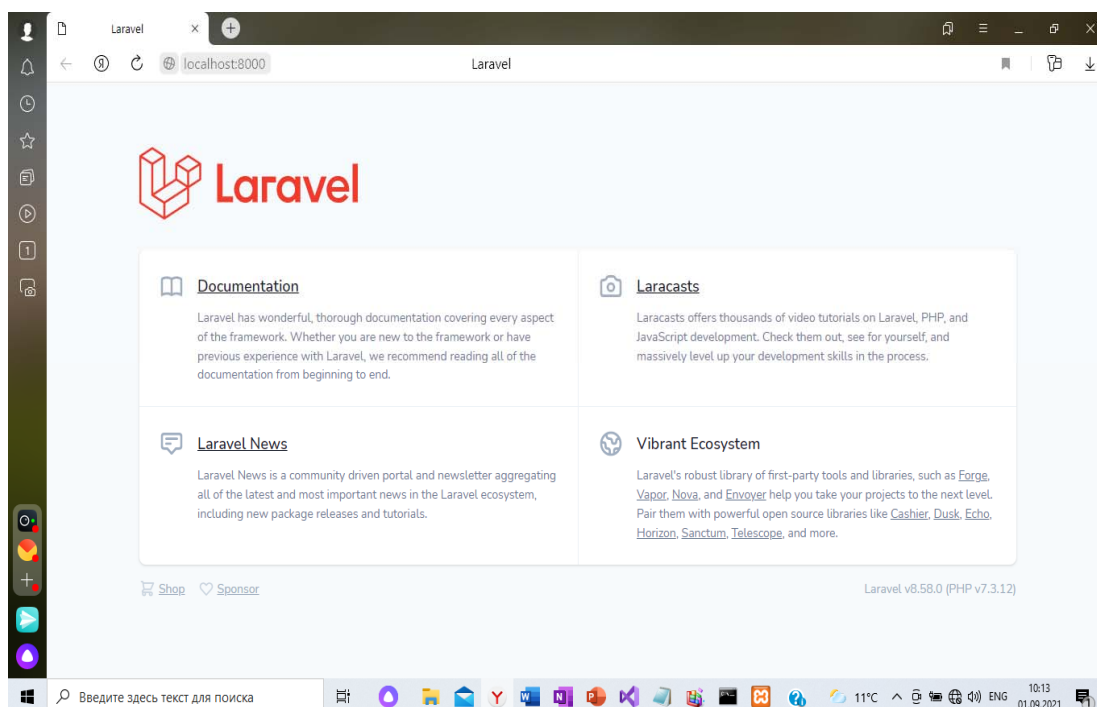


Рисунок 1.1 – Страница пустого проекта веб-сайта

Для завершения работы отладочного веб-сервера используйте «горячие клавиши» клавиатуры ($\langle Ctrl \rangle + \langle Break \rangle$ или $\langle Ctrl \rangle + \langle C \rangle$) [1; 2].

1.3 Создание контроллеров и операций

Контроллер – программный модуль, реализующий функциональность разделов сайта.

Действие – операция, выполняемая контроллером.

Веб-сайт содержит произвольное количество контроллеров (по числу разделов сайта). Контроллер содержит произвольное количество действий.

Виды контроллеров:

- контроллер-класс;
- контроллер одного действия;
- контроллер-функция [1; 3].

Создание контроллера выполняется командой
php artisan make:controller имя контроллера.

1.4 Создание веб-маршрута

Чтобы операция контроллера исполнилась при получении запроса по определенному пути, выполненного определенным *HTTP-методом*, его следует связать с этим путем и методом, создав *маршрут*.

Маршрут Laravel – это объект особого класса, содержащий следующие сведения:

- шаблонный путь – задает нужный формат путей;
- допустимый *HTTP-метод*, которым должен быть выполнен клиентский запрос;
- действие контроллера, выполняемое при совпадении шаблонного пути и допустимого метода с путём и методом, извлечёнными из запроса (если маршрут является совпавшим).

Созданные маршруты записываются в один из двух *списков*:

- 1) список *веб-маршрутов*;
- 2) список *API-маршрутов*.

Просмотр одного из списков маршрутов, в зависимости от типа полученного запроса, в поисках совпавшего выполняет *маршрутизатор*. Если ни один маршрут не совпал, выводится страница с сообщением об *ошибке 404 (запрашиваемый путь не существует)* [1; 4].

Создание веб-маршрута, связывающего шаблонный путь / («корень» сайта) и допустимый *HTTP-метод GET* с действием *ИмяДействия()* контроллера *ИмяКонтроллера*, включает этапы:

1 Открыть модуль *routes/web.php* со списком веб-маршрутов в текстовом редакторе.

2 Связать изначально созданный маршрут с действием *Имя Действия()* контроллера *ИмяКонтроллера*:

```
use App\Http\Controllers\BbsController  
Route::get('/', [ИмяКонтроллера::class, 'Имя Действия']);
```

1.5 Создание миграций

Миграция – программный *PHP-модуль*, вносящий изменения в структуру базы данных.

При *применении* миграция выполняет все описанные действия.

При *откате* миграция возвращает базу данных в состояние, существовавшее перед применением миграции.

Программные модули с миграциями хранятся в папке *database\migration*. Список всех применённых миграций в хронологическом порядке сохраняется в таблице базы данных, создаваемой перед применением самой первой миграции.

Создание миграции выполняется командой [1]

```
php artisan make:migration ИмяМиграции - --create=имятаблицы
```

1.6 Создание моделей

Модель – программный *RНР*-модуль, служащий для взаимодействия с обслуживаемой им таблицей базы данных: выборки записей, извлечения значений полей, добавления, правки и удаления записей. Модуль реализуется в виде класса. Файлы с моделями по умолчанию хранятся в папке *app\Models*.

Создание модели выполняется командой [1].

```
php artisan make:model ИмяМодели
```

1.7 Шаблоны

Шаблон – образец для генерирования страницы, отправляемой клиенту в составе ответа.

Рендеринг – процесс генерирования страницы.

Шаблонизатор – подсистема фреймворка, выполняющая рендеринг.

Для рендеринга шаблонизатору требуются шаблон и данные, которые будут выводиться на генерируемой странице. Данные оформляются в виде набора (*контекста шаблона*).

Контекст шаблона формируется контроллером в виде ассоциативного массива, который преобразуется шаблонизатором в набор обычных переменных, доступных внутри шаблона.

В *Laravel* шаблоны сохраняются в файлах с расширением *blade.php* и помещаются в папке *resources\views* или вложенных в неё папках [1].

1.8 Разграничение доступа

Разграничение доступа к данным веб-сайта, написанного с применением *Laravel*, реализуется рядом посредников и политиками.

Посредник (middleware) – программный модуль, выполняющий предварительную обработку запроса перед запуском контроллера и (или) окончательную

обработку ответа после его формирования контроллером и перед отправкой клиенту.

Политика (policy) – программный модуль, реализующий разграничение доступа к определенной модели согласно заданным правилам.

Провайдер (provider) – программный модуль, задающий режим работы и параметры подсистем фреймворка.

Посредники, политики и провайдеры реализуются в виде классов [1].

1.9 Лабораторная работа №1 «Веб-сайт коммерческих объявлений»

Цель: получить теоретические знания и практические навыки в использовании технологии php-фреймворка *Laravel*, сервера баз данных *SQLite*.

Используемые приемы и технологии: фреймворк *Laravel*, сервер баз данных *SQLite*, *HTML*, *CSS*.

Ключевые термины: фреймворк, контроллер, действие, маршрут, миграция, модель, шаблон, валидация, разграничение прав доступа.

1.9.1 Варианты заданий

Разработать веб-сайт коммерческих объявлений, размещённых на бесплатной основе:

- 1 Веб-сайт объявлений купли-продажи автомобилей.
- 2 Веб-сайт объявлений купли-продажи недвижимости.
- 3 Веб-сайт объявлений купли-продажи вело-, мототехники.
- 4 Веб-сайт объявлений купли-продажи строительных материалов.
- 5 Веб-сайт объявлений купли-продажи сельскохозяйственной техники.
- 6 Веб-сайт объявлений купли-продажи спутникового оборудования.
- 7 Веб-сайт объявлений купли-продажи вычислительной техники.
- 8 Веб-сайт объявлений купли-продажи кабельного оборудования.
- 9 Веб-сайт объявлений купли-продажи книгопечатной продукции.
- 10 Веб-сайт объявлений купли-продажи мебельной продукции.
- 11 Веб-сайт объявлений купли-продажи плодоовощной продукции.
- 12 Веб-сайт объявлений купли-продажи автозапчастей.
- 13 Веб-сайт объявлений купли-продажи вычислительной техники.
- 14 Веб-сайт объявлений купли-продажи лакокрасочной продукции.
- 15 Веб-сайт объявлений купли-продажи музыкальных товаров.
- 16 Веб-сайт объявлений купли-продажи электротоваров.
- 17 Веб-сайт объявлений купли-продажи товаров для дома и сада.

- 18 Веб-сайт объявлений купли-продажи для отдыха и путешествий.
- 19 Веб-сайт объявлений купли-продажи торгового оборудования.
- 20 Веб-сайт объявлений купли-продажи сетевого оборудования.
- 21 Веб-сайт объявлений купли-продажи товаров для физической культуры, спорта и туризма.
- 22 Веб-сайт объявлений купли-продажи теле-, радиотоваров.
- 23 Веб-сайт объявлений купли-продажи художественных товаров.
- 24 Веб-сайт объявлений купли-продажи фото-, кинотоваров.
- 25 Веб-сайт объявлений купли-продажи хозяйственных товаров.

1.9.2 Методические указания

- 1 Продумайте структуру веб-сайта. Выполните поиск графического и текстового материала для информационного контента сайта.
- 2 Выполните настройку проекта для работы с системой управления базой данных.
- 3 Создайте базу данных.
- 4 Разработайте регистрацию пользователя.
- 5 Разработайте авторизацию пользователя.
- 6 Реализуйте добавление объявления, содержащего данные:
 - описание товара;
 - изображение;
 - стоимость товара (услуги);
 - местонахождение;
 - условие доставки;
 - контактный телефон;
 - адрес электронной почты.
- 7 Реализуйте вывод объявлений для просмотра.
- 8 Реализуйте правку объявлений.
- 9 Реализуйте удаление объявлений.
- 10 Реализуйте поиск объявлений.
- 11 Создайте разграничение прав.
- 12 Проведите отладку веб-сайта.

1.9.3 Контрольные вопросы

- 1 Какое существует определение контроллера php-фреймворка *Laravel*?
- 2 Каким образом определяется связывание посредников с контроллером-классом php-фреймворка *Laravel*?

- 3 Какое существует определение маршрута php-фреймворка *Laravel*?
- 4 Какие существуют условия самостоятельного неявного внедрения моделей php-фреймворком *Laravel*?
- 5 Какое существует определение миграции php-фреймворка *Laravel*?

1.10 Лабораторная работа №2 «Веб-сайт фирмы, осуществляющей продажу товаров»

Цель: получить теоретические знания и практические навыки в разработке веб-сайта с использованием php-фреймворка *Laravel*, сервера баз данных *MySQL*.

Используемые приемы и технологии: фреймворк *Laravel*, сервер баз данных *MySQL*, *HTML*, *CSS*.

Ключевые термины: архитектура *MVC*, структура веб-сайта, макет веб-сайта, динамическое прототипирование, сервер баз данных, запросы, хостинг.

1.10.1 Варианты заданий

Разработать веб-сайт фирмы, выполняющей продажу товаров:

- 1 Веб-сайт по продаже садовых растений для компании «Урожай».
- 2 Веб-сайт по продаже легковых автомобилей для компании «Авто».
- 3 Веб-сайт по продаже вело-, мототехники для компании «Мототехника».
- 4 Веб-сайт по продаже музыкальных инструментов для компании «Муза».
- 5 Веб-сайт по продаже электротоваров для компании «Электрон».
- 6 Веб-сайт по продаже спутникового оборудования для компании «Спутник».
- 7 Веб-сайт по продаже строительных материалов для компании «Стройматериалы».
- 8 Веб-сайт по продаже сельскохозяйственной техники для компании «Нива».
- 9 Веб-сайт по продаже лакокрасочной продукции для компании «Лак».
- 10 Веб-сайт по продаже вычислительной техники для компании «Компьютер».
- 11 Веб-сайт по продаже спортивных товаров для компании «Чемпион».
- 12 Веб-сайт по продаже теле-, радиотоваров для компании «Телеком».
- 13 Веб-сайт по продаже художественных товаров для компании «Авангард».
- 14 Веб-сайт по продаже фото-, кинотоваров для компании «Объектив».
- 15 Веб-сайт по продаже хозяйственных товаров для компании «Хозтовары».

- 16 Веб-сайт по продаже мебельной продукции для компании «Мебель».
- 17 Веб-сайт по продаже книгопечатной продукции для компании «Книга».
- 18 Веб-сайт по продаже парфюмерной продукции для компании «Парфюм».
- 19 Веб-сайт по продаже канцелярских принадлежностей для компании «Карандаш».
- 20 Веб-сайт по продаже средств связи для компании «Аплинк».
- 21 Веб-сайт по продаже кабельного оборудования для компании «Кабель».
- 22 Веб-сайт по продаже ювелирных изделий для компании «Изумруд».
- 23 Веб-сайт по продаже плодовоовощной продукции для компании «Фрукты-овощи».
- 24 Веб-сайт по продаже автозапчастей для компании «Автозапчасти».
- 25 Веб-сайт по продаже недвижимости для компании «Дом».

1.10.2 Методические указания

- 1 Продумайте структуру веб-сайта. Выполните поиск графического и текстового материала для информационного контента веб-сайта.
- 2 Разработайте структуру веб-сайта.
- 3 Создайте макет веб-сайта. Выполните динамическое прототипирование.
- 2 Выполните настройку проекта для работы с системой управления базой данных.
- 3 Создайте базу данных.
- 4 Реализуйте функциональности пользовательской части веб-сайта:
 - регистрация пользователя;
 - авторизация пользователя;
 - каталог товаров;
 - карточка товара;
 - поиск товаров;
 - корзина покупателя;
 - оформление заказа;
 - расчет прогнозных данных.
- 5 Создайте админ-панель.
- 6 Проведите отладку веб-сайта.

1.10.3 Контрольные вопросы

- 1 Какое существует определение модели php-фреймворка *Laravel*?
- 2 Какой формат имеет команда создания модели php-фреймворка *Laravel*?

- 3 Какое существует определение шаблона php-фреймворка *Laravel*?
- 4 Какие существуют настройки шаблонизатора php-фреймворка *Laravel*?
- 5 Какие существуют директивы шаблонизатора php-фреймворка *Laravel*?

2 ФРЕЙМВОРК ANGULAR

Angular – открытый фреймворк для разработки реактивных динамических одностраничных веб-приложений, написанных на языке программирования *TypeScript*.

2.1 Установка программного обеспечения

Для разработки веб-приложений с использованием фреймворка *Angular* необходимо установить программное обеспечение:

- 1 сервер *Node.js* и пакетный менеджер *npm*;
- 2 *Angular CLI* – интерфейс командной строки;
- 3 Редактор кода веб-приложений.

Node.js – сервер для запуска и выполнения серверных приложений. Загрузите дистрибутив с сайта <https://nodejs.org/ru>. Выполните установку *Node.js* (рисунок 2.1).

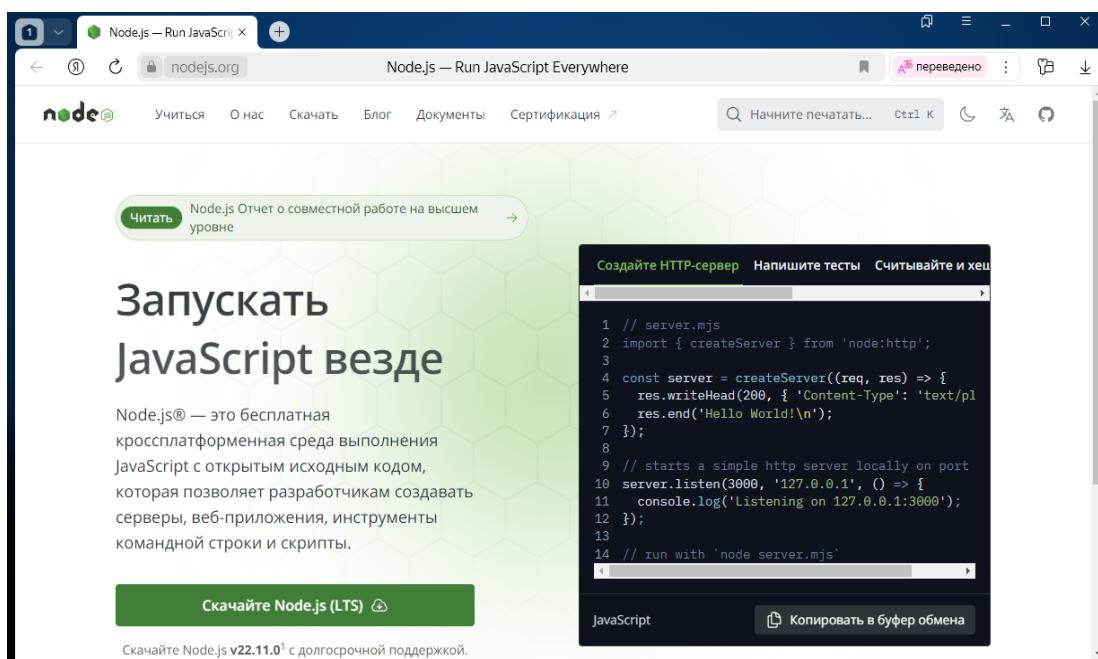


Рисунок 2.1 – Страница веб-сайта для скачивания дистрибутива сервера Node.js

Angular CLI – инструмент создания и управления пакетами *Angular* в ходе разработки (интерфейс командной строки). Для установки *Angular CLI* введите команду в командной строке `npm install -g @angular/cli`.

Visual Studio Code – бесплатный кроссплатформенный редактор с открытым кодом (рисунок 2.2). Редактор доступен по адресу <https://code.visualstudio.com/download>. [5; 6].

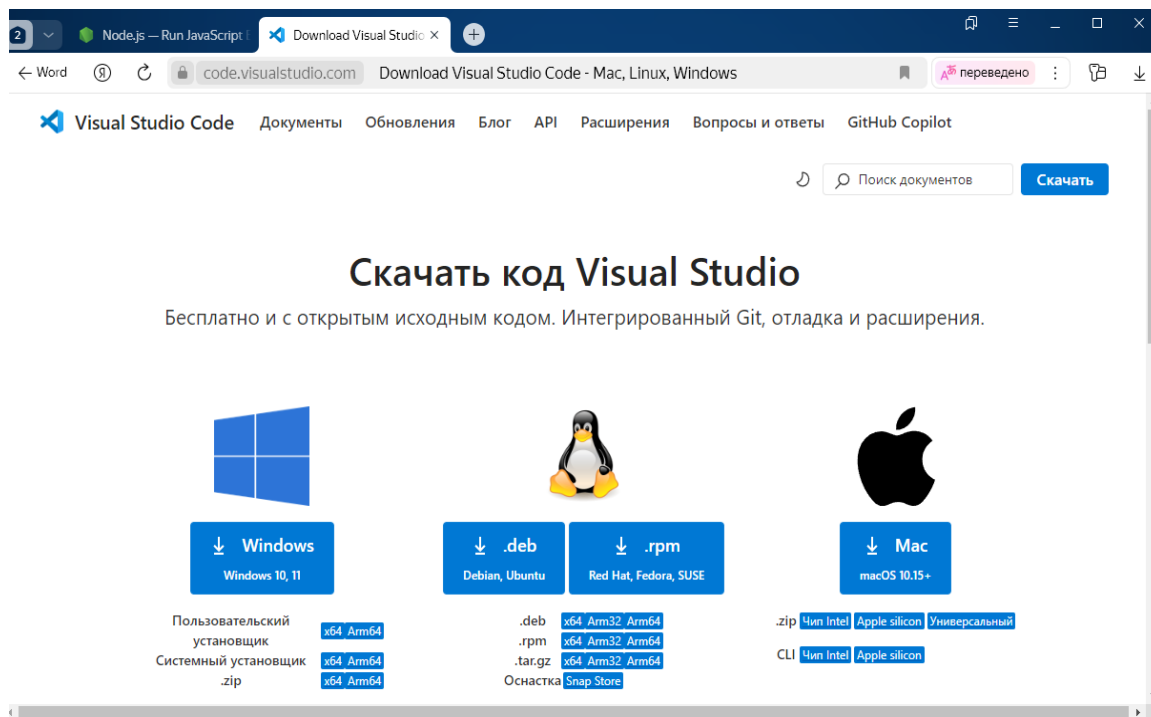


Рисунок 2.2 – Страница веб-сайта для скачивания дистрибутива *Visual Studio Code*

2.2 Создание и запуск пустого проекта веб-сайта

1 Создать папку проекта *Lab1*.

2 Перейти в папку *Lab1* в режиме командной строки `cd C:\Projects\Angular\Lab1`.

3 Создать *Angular* проект *Prog1*, выполнив команду `ng new Prog1`.

4 В ходе создания проекта ответить на вопросы:

- Хотели бы добавить маршрутизацию *Angular*? *N*.
- Какую технологию стилей будете использовать? *CSS*.

5 После создания проекта *Prog1* ознакомиться с содержимым папки проекта *Prog1* и подкаталогов *SRC* и *app* (рисунки 2.3–2.5) [5; 7].

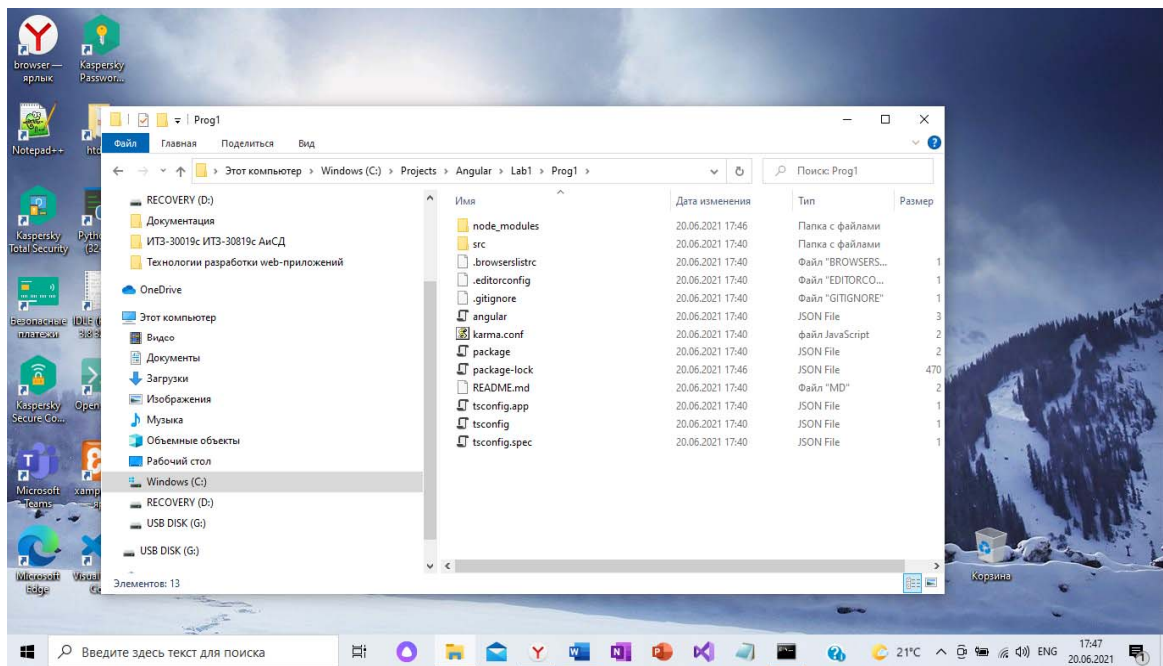


Рисунок 2.3 – Содержимое папки *Prog1*

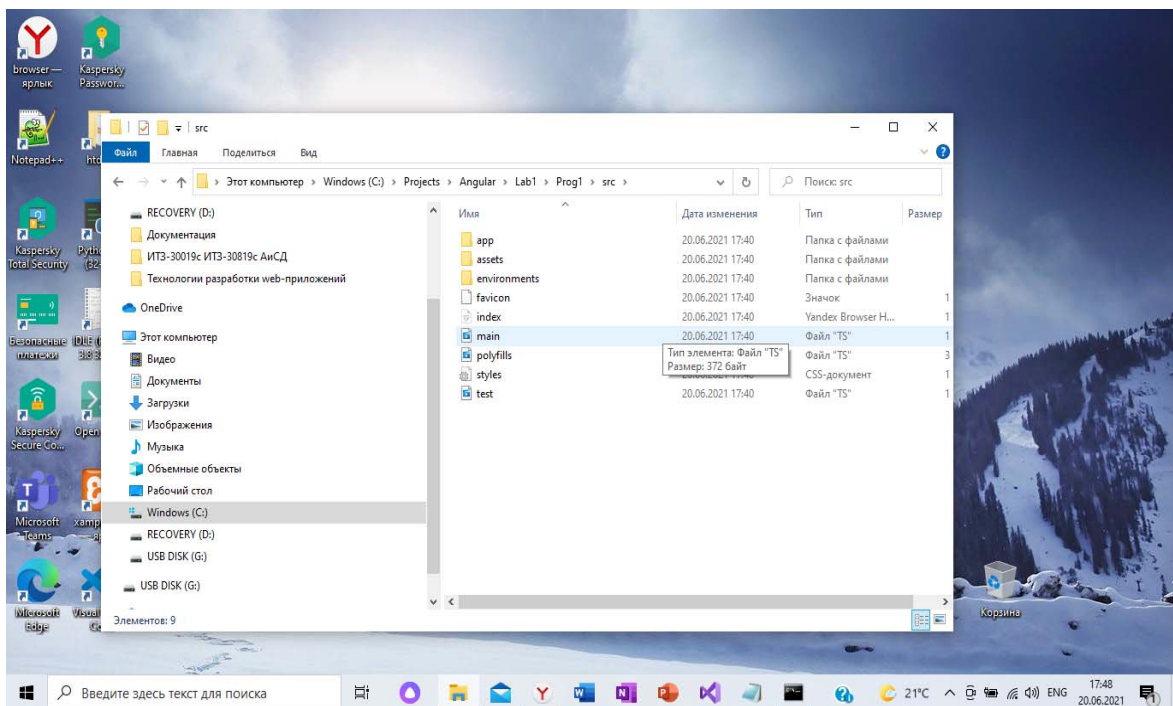


Рисунок 2.4 – Содержимое подкаталога *SRC*

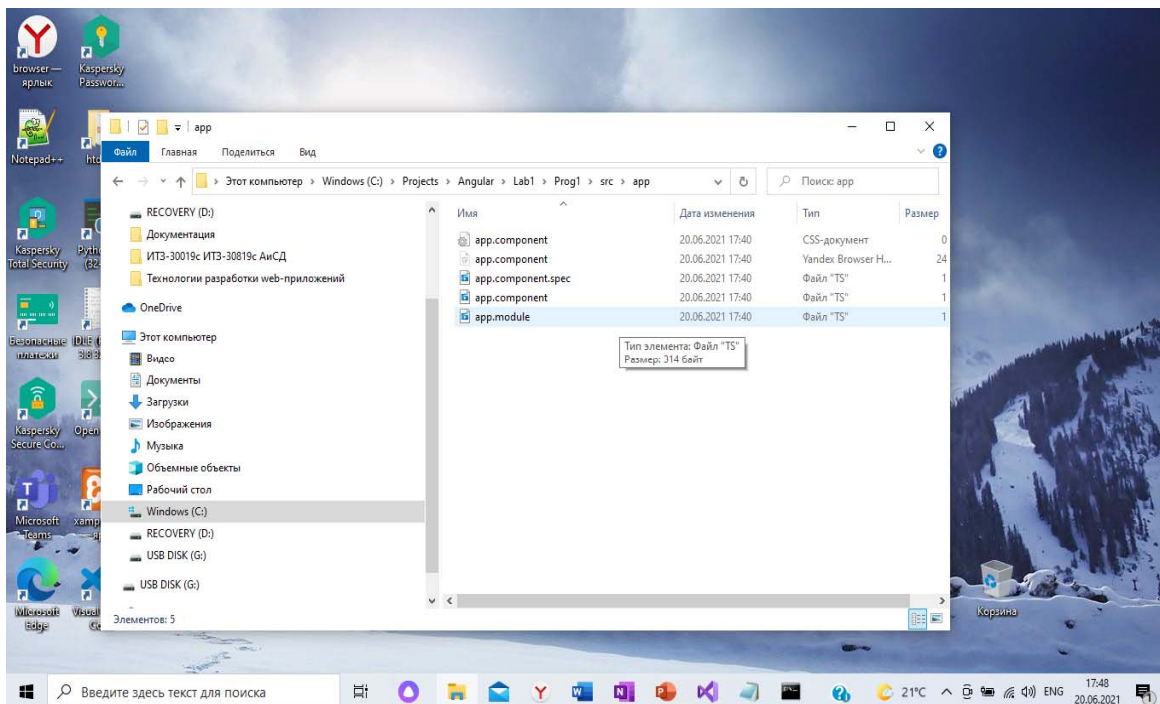


Рисунок 2.5 – Содержимое подкаталога *app*

6 Подключить к проекту библиотеку *Bootstrap*. Открыть в редакторе *Microsoft Visual Studio Code* файл `.\Prog1\package.json` и в раздел *dependencies* добавить `"bootstrap": "5.0.1"`.

7 Обработать файл *package.json*, выполнив команду `npm install`.

8 Запустить сервер и проект, выполнив команду `ng serve --port 2000 --open [5; 8]`.

Вид веб-страницы пустого проекта *Angular* представлен на рисунке 2.6.

2.3 Компоненты в Angular

Компонент (Angular component) – обособленная часть функциональности веб-приложения, включающая логику, *HTML*-шаблон и *CSS*-стили.

Компоненты *Angular* создаются в папке *app*. Компоненту в проекте соответствует три файла:

1 `[Имя_компонента].component.ts` – хранится функциональность компонента.

2 `[Имя_компонента].component.html` – хранится верстка компонента.

3 `[Имя_компонента].component.css` – хранится оформление компонента.

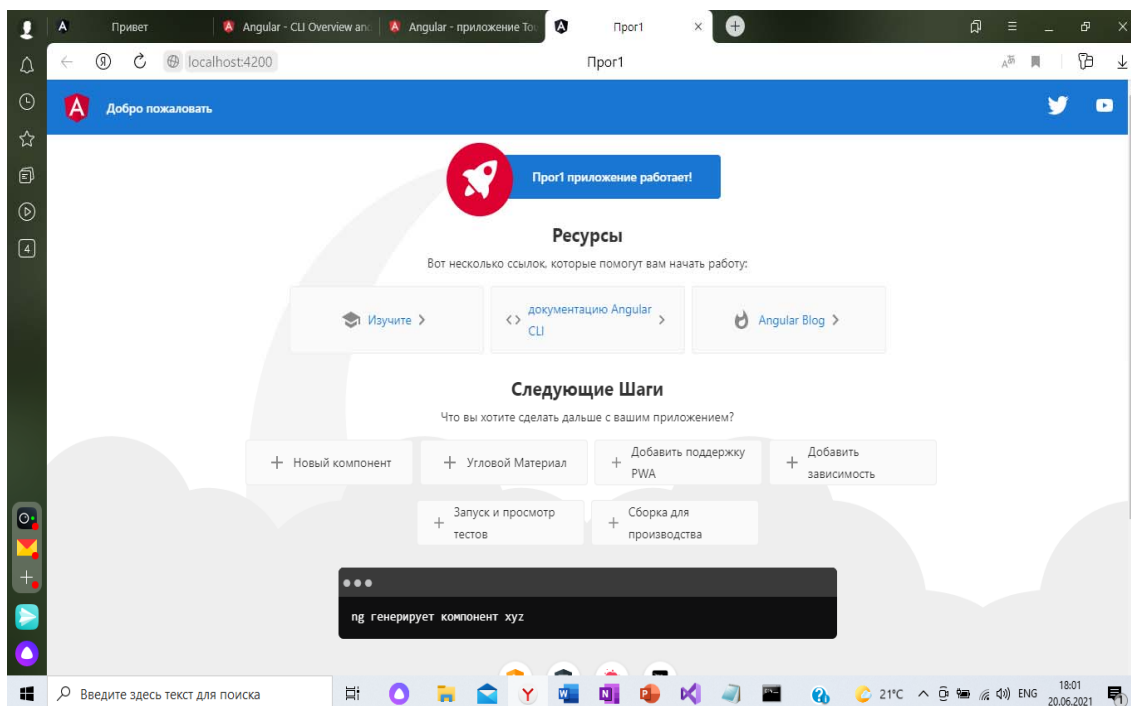


Рисунок 2.6 – Веб-страница пустого проекта *Angular*

Структура ключевого файла компонента *[Имя_компонента].component.ts* содержит разделы:

- импортирование;
- декораторы;
- класс компонента.

Создание компонента в *Angular-приложении* производится двумя способами:

- 1 Ручной.
- 2 Автоматический.

Ручной способ предусматривает создание файлов, составляющих компонент в отдельной подпапке папки *src\app* проекта с последующей регистрацией компонента в модуле, вручную.

Автоматический способ предусматривает создание компонента с помощью команды *angular-CLI*

ng generate component Имя_компонента [5; 9].

2.4 Декораторы в *Angular*

Декоратор Angular – функция, с помощью которой прикрепляются метаданные к объявлению класса, методу, средству доступа, свойству или параметру.

Для применения декоратора используется форма *@expression*, где *expression* – это имя декоратора.

Для включения *Angular* декоратора, требуется добавить *experimentalDecorators* в файл *tsconfig.json*. Команда *ng new* делает это автоматически.

Angular использует декораторы для предоставления метаданных классу, методу или свойству. Декоратор передает конфигурационную информацию в свойствах. Декоратор *@Component* включает свойства:

- selector;
- templateUrl;
- styleUrls.

Фабрика декораторов – функция, возвращающая выражение, которая будет вызвана декоратором во время исполнения программы [5; 10].

2.5 Сервисы в Angular

Сервисы в Angular – классы, которые выполняют специфические задачи, например, логгирование, работу с данными. В отличие от компонентов и директив сервисы не работают с представлениями, то есть с *HTML*-разметкой, и не оказывают на неё прямого влияния.

Стандартные задачи сервисов:

1 Предоставление данных приложению. Сервис может сам хранить данные в памяти либо для получения данных может обращаться к какому-нибудь источнику данных, например, к серверу.

2 Представление канала взаимодействия между отдельными компонентами приложения.

3 Инкапсуляция бизнес-логики, различных вычислительных задач, задач по логгированию, которые лучше выносить из компонентов [5; 11].

Чтобы задействовать сервис в компоненте, необходимо импортировать, добавить в коллекцию *providers* компонента. После этого можно задействовать встроенный в *Angular* механизм *dependency injection* и получить объект сервиса в конструкторе компонента и использовать по необходимости.

2.6 Маршрутизация в Angular

Маршрутизация в Angular – сопоставление запросов к приложению с определёнными ресурсами внутри приложения. Она обеспечивает навигацию между представлениями приложения.

За организацию маршрутизации отвечает модуль *RouterModule* библиотеки *@angular/router*. *URL* организуются в специальные модули и определяются для каждого отдельного модуля приложения.

Для указания маршрута применяется параметр *path*. При загрузке приложения *Router* выполняет начальную навигацию по текущему *URL*, который стоит в адресной строке браузера.

При запросе маршрутизатор ищет первый подходящий маршрут в порядке их определения. Поэтому обработка любого *URL* всегда должна определяться в самом конце [5; 11].

2.7 Связывание данных в Angular

Связывание данных – синхронизация в реальном времени визуальных элементов компонента с данными, хранимыми в классе компонента.

Существуют два типа связывания в *Angular*:

- однонаправленное (*one-way data binding*);
- двунаправленное (*two-way data binding*).

Однонаправленное связывание (one-way data binding) в Angular – это одностороннее взаимодействие, при котором поток данных идёт либо от класса компонента к представлению, либо от представления к классу компонента. Оно позволяет получить доступ к свойству класса компонента в соответствующем шаблоне представления и достигается с помощью интерполяции и привязки свойств в *Angular*. Поток данных от шаблона представления к соответствующему классу компонента осуществляется через привязку событий. При однонаправленном потоке данных нет необходимости отслеживать изменения состояния, потому что изменения данных приведут к полному изменению представления. Для привязки данных от класса компонента к представлению используются два типа скобок: «[]» или «{{ }}», а от представления к классу компонента – ().

Двунаправленное связывание (two-way data binding) в Angular – автоматическая синхронизация данных между свойством класса компонента и элементом ввода (например, полем ввода) в шаблоне. Двусторонняя привязка позволяет динамически менять значения на одном конце привязки при изменениях на другом конце. Как правило, она применяется при работе с элементами ввода. Для включения двунаправленного связывания требуется выполнить действия:

1 В файле компонента импортировать *FormsModule*.

2 Добавить импорт в свойство *imports* декоратора *@Component*.

Для создания двусторонней привязки в элементе ввода нужно написать специальную директиву *[(ngModel)]*, значением которой следует указать свойство класса [5; 11].

2.8 Директивы в Angular

Angular директивы – инструменты, расширяющие *HTML*-код настраиваемым поведением, являющиеся маркерами на элементах *DOM*, которые сообщают *Angular* что требуется сделать с элементом или его дочерними элементами [5; 11].

Директивы фреймворка *Angular*, применяемые при связывании, приводятся в таблице 2.1.

Таблица 2.1 – Встроенные директивы *Angular*

Наименование	Описание
<i>ngClass</i>	Определение наборов классов, которые будут применяться к элементу. Директива позволяет динамически добавлять или удалять классы <i>CSS</i> на основе свойств компонента или состояния приложения
<i>ngStyle</i>	Задание наборов стилей, которые применяются к элементу. В качестве значения директива принимает <i>js</i> -объект, в котором ключи – названия свойств <i>CSS</i>
<i>ngIf</i>	Условное включение или удаление элемента на основе представленного выражения. Если выражение оценивается как <i>true</i> , элемент включается в <i>DOM</i> , если как <i>false</i> – удаляется из <i>DOM</i>
<i>ngFor</i>	Выполнение итерации по коллекции и создание экземпляра шаблона один раз для каждого элемента в коллекции
<i>ngSwitch</i> ,	Отображение одного или нескольких элементов <i>DOM</i> на основе заранее определённого условия
<i>ngSwitchCase</i>	Добавление своего элемента в <i>DOM</i> , когда его связанное значение равно значению <i>switch</i> , и удаляет его связанное значение, когда оно не равно значению <i>switch</i>
<i>ngSwitchDefault</i>	Создание вида, который добавляется к родительскому <i>NgSwitch</i> , когда ни одно из выражений случаев не соответствует выражению переключения
<i>ngtemplateOutlet</i>	Выполнение вставки содержимого шаблона, определённого с помощью <i>ng-template</i> , в любой другой элемент шаблона. Это позволяет создавать динамически изменяемые пользовательские интерфейсы

2.9 Лабораторная работа №3

«Веб-приложение “Выдающиеся деятели России”»

Цель: получить теоретические знания и практические навыки в разработке веб-сайта с использованием фреймворка *Angular*.

Используемые приемы и технологии: фреймворк *Angular*, сервер *Node.js*, *Angular CLI*, пакетный менеджер *npm*, язык программирования *TypeScript*, *HTML*, *CSS*, одностраничное веб-приложение.

Ключевые термины: *Angular*, *TypeScript*, *HTML-документ*, каскадные таблицы стилей, одностраничный веб-сайт.

2.9.1 Варианты заданий

Разработать *Angular-приложение*, являющееся интерактивным каталогом выдающихся деятелей России. Веб-приложение содержит фотогалерею с возможностью поиска. В поле ввода поиска можно вводить фамилию человека. Веб-приложение будет искать и выводить результат поиска в кратком предварительном представлении. При необходимости предусмотрена возможность выдачи полной информации с подробным описанием заслуг и графическим изображением.

- 1 Выдающиеся драматурги России.
- 2 Выдающиеся актеры кино, театра России.
- 3 Выдающиеся космонавты России.
- 4 Выдающиеся кинокритики России.
- 5 Выдающиеся кинорежиссеры России.
- 6 Выдающиеся бизнесмены России.
- 7 Депутаты Государственной Думы России.
- 8 Выдающиеся врачи России.
- 9 Выдающиеся полководцы России.
- 10 Почетные граждане г. Кургана.
- 11 Лауреаты Нобелевской премии.
- 12 Выдающиеся спортсмены России.
- 13 Выдающиеся музыканты России.
- 14 Выдающиеся художники России.
- 15 Выдающиеся политические деятели России.
- 16 Выдающиеся артисты эстрады России.
- 17 Выдающиеся писатели России.
- 18 Выдающиеся поэты России.
- 19 Выдающиеся певцы России.
- 20 Выдающиеся композиторы России.
- 21 Выдающиеся учёные России.
- 22 Выдающиеся конструкторы России.
- 23 Выдающиеся изобретатели России.
- 24 Выдающиеся путешественники России.

25 Выдающиеся летчики России.

2.9.2 Методические указания

- 1 Продумайте структуру веб-сайта. Выполните поиск графического и текстового материала для информационного контента сайта.
- 2 Разработайте структуру веб-сайта.
- 3 Создайте макет веб-сайта. Выполните динамическое прототипирование.
- 4 Реализуйте функциональности пользовательской части веб-сайта:
 - реализация фотогалереи выдающихся деятелей России;
 - реализация поиска по имени выдающегося деятеля России;
 - реализация вывода краткой информации о выдающемся деятеле России;
 - реализация вывода полной информации о выдающемся деятеле России.
- 5 Проведите отладку веб-сайта.

2.9.3 Контрольные вопросы

- 1 Какое существует определение фреймворка *Angular*?
- 2 Какое существует определение одностраничного веб-приложения (*SPA*)?
- 3 Какие существуют преимущества одностраничных веб-приложений?
- 4 Какие папки и файлы включает структура *Angular-проекта*?
- 5 Какие используются команды для создания и запуска *Angular-проекта*?

2.10 Лабораторная работа №4 «Веб-приложение “Список покупок”»

Цель: получить теоретические знания и практические навыки в разработке веб-сайта с использованием фреймворка *Angular*.

Используемые приемы и технологии: фреймворк *Angular*, язык программирования *TypeScript*, *HTML*, *CSS*, компонент *Angular*, модуль *Angular*, модель данных.

Ключевые термины: *Angular*, *TypeScript*, компонент, темплейт, стили компонента, программная логика компонента, декоратор, свойства декоратора.

2.10.1 Варианты заданий

Разработать веб-приложение, реализующее список покупок товаров с функциями добавления, просмотра, поиска и удаления приобретенных коммерческих товаров.

- 1 Список покупки хозяйственных товаров.
- 2 Список покупки фото-, кинотоваров.
- 3 Список покупки художественных товаров.
- 4 Список покупки теле-, радиотоваров.
- 5 Список покупки товаров для физической культуры, спорта и туризма.
- 6 Список покупки сетевого оборудования.
- 7 Список покупки торгового оборудования.
- 8 Список покупки для отдыха и путешествий.
- 9 Список покупки товаров для дома и сада.
- 10 Список покупки электротоваров.
- 11 Список покупки музыкальных товаров.
- 12 Список покупки лакокрасочной продукции.
- 13 Список покупки плодоовощной продукции.
- 14 Список покупки автозапчастей.
- 15 Список покупки вычислительной техники.
- 16 Список покупки мебельной продукции.
- 17 Список покупки книгопечатной продукции.
- 18 Список покупки кабельного оборудования.
- 19 Список покупки вычислительной техники.
- 20 Список покупки спутникового оборудования.
- 21 Список покупки сельскохозяйственной техники.
- 22 Список покупки строительных материалов.
- 23 Список покупки вело-, мототехники.
- 24 Список покупки недвижимости.
- 25 Список покупки автомобилей.

2.10.2 Методические указания

1 Продумайте структуру веб-сайта. Выполните поиск графического и текстового материала для информационного контента сайта.

2 Разработайте структуру веб-сайта.

3 Создайте макет веб-сайта. Выполните динамическое прототипирование.

4 Реализуйте функциональности пользовательской части веб-сайта:

- реализация вывода списка покупок;

- реализация добавления купленного товара;
- реализация поиска товара в списке;
- реализация удаления товара из списка.

5 Проведите отладку веб-сайта.

2.10.3 Контрольные вопросы

- 1 Что понимается под компонентом фреймворка *Angular*?
- 2 Какая существует файловая структура компонента *Angular*?
- 3 Что понимается под декоратором фреймворка *Angular*?
- 4 Какие свойства включает декоратор фреймворка *Angular*?
- 5 Что понимается под метамодулем *Angular*?

2.11 Лабораторная работа №5 «Веб-приложение “Трекер привычек”»

Фреймворк *Angular* и модуль *Angular Material* предоставляют набор бесплатных компонентов, стилей и поведений. Модуль *Angular Material* помогает создавать высококачественные приложения с пользовательским интерфейсом на основе фреймворка *Angular*, следуя спецификациям *Material Design*.

Цель: получить теоретические знания и практические навыки в разработке веб-сайта с использованием фреймворка *Angular*, модуля *Angular Material*.

Используемые приемы и технологии: фреймворк *Angular*, модуль *Angular Material*, модули *Angular CDK*, *Angular Animations*, темы *Angular Material*, спецификация *Material Design*, *HTML*, *SCSS*.

Ключевые термины: *Angular*, *Angular Material*, *TypeScript*, иконки, компонент, темплейт, маршрутизация, реактивная форма, программная логика компонента, декоратор, свойства декоратора.

2.11.1 Варианты заданий

Разработать веб-приложение для отслеживания привычек, содержащее соответствующие записи, которые можно будет добавлять, редактировать и удалять.

1 Личные привычки.

Личные привычки связаны с Вашей повседневной деятельностью:

- посмотрите обучающее видео о программах, которые Вы используете каждую неделю;
- анализируйте отрицательные эмоции;

- просыпайтесь в одно и то же фиксированное время;
- ложитесь спать вовремя;
- принимайте душ каждый день;
- правильно завтракайте;
- читайте по 10 страниц книги в день;
- не пользуйтесь электронными гаджетами за час до сна;
- используйте трекер привычек;
- спите не менее 8 часов;
- выполняйте действия для продвижения к поставленным целям;
- тренируйте свой мозг с помощью специальных тренажеров.

2 Привычки для построения правильных отношений.

Отслеживание своих привычек в отношениях помогает Вам понять, как Вы проводите время в семье, с друзьями и другими важными людьми в Вашей жизни:

- звоните членам семьи один раз в день;
- напишите сообщение другу;
- сходите на встречу;
- не используйте свой смартфон в компании других людей;
- делайте комплименты другим;
- участвуйте в группах поддержки, сообществах по интересам;
- сделайте приятное другу;
- участвуйте в волонтерстве.

3 Привычки для социальных сетей.

Отслеживание своего использования социальных сетей и взаимодействия с другими людьми в социальных сетях:

- проводите в социальных сетях не более определенного количества времени;
- разместите фотографию;
- не используйте социальные сети в своей работе;
- напишите жизнеутверждающий комментарий под публикацией или фотографией.

4 Привычки для ведения блога.

Отслеживание промежуточных шагов создания блога:

- напишите пару пробелов в блоге;
- напишите черновик записи;
- опубликуйте новую статью;
- оптимизируйте старый пост;

- ответьте на комментарии читателей;
- ответьте на электронные письма подписчиков;
- посмотрите видеоурок по созданию контента;
- работайте над созданием индивидуального курса;
- напишите один абзац текста в день;
- присоединяйтесь к сообществу блогеров.

5 Рабочие привычки.

Отслеживание рабочих привычек:

- не сплетничайте с коллегами на работе;
- выполняйте одно действие, которое приводит к завершению решения текущей задачи;
 - выполняйте уборку рабочего места в конце рабочего дня;
 - сделайте несколько деловых звонков;
 - ответьте на деловые письма по электронной почте;
 - сходите на встречу;
 - просматривайте свои заметки, повторяйте информацию;
 - представьте идею на встрече;
 - выделите пять минут в день на размышления об улучшениях в Ваших процессах;
 - напишите аннотации к каждому показанному видео по специальности (особенно к тем, которые можно применить).

6 Привычки для здоровья и долголетия.

Отслеживание привычек заботы о здоровье и долголетию:

- ешьте определенное количество фруктов в день;
- ешьте определенное количество овощей в день;
- не ешьте нездоровую пищу);
- не ешьте больше 30 граммов сахара и простых углеводов в день;
- пейте по 500 мл минеральной воды каждые 3 дня;
- не употребляйте алкогольные напитки;
- избегайте перекусов, планируйте утреннее питание и придерживайтесь этого плана;
 - не ешьте больше N калорий;
 - не ешьте более X углеводов Y белков Z жиров;
 - выпивайте не менее 1 литра чистой воды на каждые 30 кг массы тела;
 - делайте физические упражнения;
 - гуляйте на свежем воздухе;
 - посещайте спортзал;

- ешьте только домашнюю еду, не покупайте готовые блюда в супермаркетах;

- умойтесь, почистите зубы;
- откажитесь от курения;
- сделайте 10 000 шагов.

7 Привычки по уборке.

Отслеживание привычек по уборке:

- не оставляйте грязную посуду на следующий день;
- заправьте кровать;
- уберите всё со стола, очистите рабочие поверхности;
- разложите одежду по местам;
- протрите пыль в квартире;
- помойте пол в квартире;
- вынесите мусор;
- удалите ненужные файлы с жесткого диска Вашего ПК;
- проверяйте список своих подписок на платформах один раз в неделю. Исключите из них тех, на кого подписались под влиянием эмоций;
- расставьте вещи в квартире на свои места.

8 Привычки управления личными финансами.

Отслеживание привычек управления финансами:

- создайте финансовую подушку безопасности;
- откладывайте 10% дохода на финансовый счёт сбережения;
- записывайте доходы и расходы каждый день;
- исключите покупки товаров, без которых Вы можете обойтись;
- не берите крупную сумму денег при посещении магазина;
- ищите дополнительный заработок к основной работе;
- повышайте уровень знаний, умений и компетенций.

9 Привычки для саморазвития и карьеры.

Отслеживание привычек для саморазвития, самообладания и успеха в продвижении по карьерной лестнице:

- уделите 15 минут в день изучению новых навыков;
- читайте профессиональные книги и литературу по самопомощи;
- тренируйте память и пополняйте словарный запас;
- выделите 15 минут в день на составление плана расстановки приоритетов задач и распределению времени и усилий;
- выделите 15 минут в день на пополнение своего портфолио и обновление своего веб-сайта;

- продвигайте себя как специалиста, даже если на текущей должности вас все устраивает;
 - говорите о деле при каждой встрече с коллегами и друзьями.
- 10 Привычки для гармонии с собой и окружением.
- Отслеживание привычек для гармонии с собой и окружением (психологическое спокойствие и хорошие отношения с окружающей средой):
- находите хорошее и улыбайтесь каждый день;
 - уделите две минуты времени, чтобы встать перед зеркалом и улыбаться своему отражению;
 - выделите время на медитацию каждый день;
 - поблагодарите себя за 5 минут до сна, чтобы научиться сосредотачиваться на хорошем;
 - напишите о своих чувствах в дневнике;
 - привыкайте к жалобам, сплетням и критике;
 - не откладывайте мелкие домашние дела, чтобы они не накапливались [12].

2.11.2 Методические указания

1 Продумайте структуру веб-сайта. Выполните поиск графического и текстового материала для информационного контента сайта.

2 Разработайте структуру веб-сайта.

3 Создайте макет веб-сайта. Выполните динамическое прототипирование.

4 Создайте *Angular* проект. Добавьте маршрутизацию *Angular* и выберите формат таблицы стилей SCSS.

Would you like to add Angular routing? Y.

Which stylesheet format would you like to use? SCSS.

5 Установите *Angular Material* и используйте ряд его компонентов для создания красивого *UI*.

6 Включите в шаблон *UI* структурные директивы *Angular* для показа и сокрытия элементов на основе состояния компонента.

7 Создайте реактивную форму для получения входных данных.

8 Реализуйте функциональности пользовательской части веб-сайта:

- вывод списка привычек;
- добавление привычки в список;
- поиск привычки;
- удаление привычки.

5 Проведите отладку веб-сайта.

2.11.3 Контрольные вопросы

- 1 Какое предназначение имеет система *Material Design*?
- 2 Какие существуют принципы системы *Material Design*?
- 3 Какое определение можно дать *Angular Material*?
- 4 Какая команда применяется для создания проекта *Angular Material*?
- 5 Какая команда используется для установки *Angular* модулей *CDK*?

2.12 Лабораторная работа №6 «Веб-сайт электронной коммерции»

Цель: получить теоретические знания и практические навыки в разработке веб-сайта с использованием фреймворка *Angular*, системы управления базой данных (*RDBMS*) *MySQL*.

Используемые приемы и технологии: фреймворк *Angular*, СУБД *MySQL*, *REST API*, *SQL*, стек *MEAN*, клиентская часть, серверная часть, язык программирования *TypeScript*, *HTML*, *CSS*, компонент *Angular*, модуль *Angular*.

Ключевые термины: *Angular*, *TypeScript*, база данных, концептуальная модель, схема данных, таблица, внешний ключ, первичный ключ, нормализация.

2.12.1 Варианты заданий

Разработать веб-сайт фирмы, выполняющей продажу товаров:

- 1 Веб-сайт фирмы «Сад» по продаже садовых растений.
- 2 Веб-сайт фирмы «Авто» по продаже легковых автомобилей.
- 3 Веб-сайт фирмы «Мототехника» по продаже вело-, мототехники.
- 4 Веб-сайт фирмы «Муза» по продаже музыкальных инструментов.
- 5 Веб-сайт фирмы «Электрон» по продаже электротоваров.
- 6 Веб-сайт фирмы «Спорттовары» по продаже товаров для физической культуры, спорта и туризма.
- 7 Веб-сайт фирмы «Волна» по продаже теле-, радиотоваров.
- 8 Веб-сайт «Графика» по продаже художественных товаров.
- 9 Веб-сайт «Фотостудия» по продаже фото-, кинотоваров.
- 10 Веб-сайт «Хозтовары» по продаже хозяйственных товаров
- 11 Веб-сайт «Спутник» по продаже спутникового оборудования.
- 12 Веб-сайт «Стройматериал» по продаже строительных материалов.
- 13 Веб-сайт «Нива» по продаже сельскохозяйственной техники.
- 14 Веб-сайт «Лак» по продаже лакокрасочной продукции.
- 15 Веб-сайт «Компьютер» по продаже вычислительной техники.

- 16 Веб-сайт «Мебель» по продаже мебельной продукции.
- 17 Веб-сайт «Книга» по продаже книгопечатной продукции.
- 18 Веб-сайт «Парфюм» по продаже парфюмерной продукции.
- 19 Веб-сайт «Карандаш» по продаже канцелярских принадлежностей.
- 20 Веб-сайт «Радио» по продаже средств связи.
- 21 Веб-сайт «Кабель» по продаже кабельного оборудования.
- 22 Веб-сайт «Изумруд» по продаже ювелирных изделий.
- 23 Веб-сайт «Фрукты-овощи» по продаже плодовоовощной продукции.
- 24 Веб-сайт «Автозапчасти» по продаже автозапчастей.
- 25 Веб-сайт «Дом» по продаже недвижимости.

2.12.2 Методические указания

- 1 Продумайте структуру веб-сайта. Выполните поиск графического и текстового материала для информационного контента веб-сайта.
- 2 Разработайте структуру веб-сайта.
- 3 Создайте макет веб-сайта. Выполните динамическое прототипирование.
- 2 Выполните настройку проекта для работы с системой управления базой данных.
- 3 Реализуйте базу данных.
- 4 Реализуйте функциональности пользовательской части веб-сайта:
 - регистрация пользователя;
 - авторизация пользователя;
 - каталог товаров;
 - карточка товара;
 - поиск товаров;
 - корзина покупателя;
 - оформление заказа;
 - расчет прогнозных данных.
- 5 Создайте админ-панель.
- 6 Проведите отладку веб-сайта.

2.12.3 Контрольные вопросы

- 1 Какое определение можно дать *СУБД*?
- 2 Каким образом можно создать базу данных *MySQL*?
- 3 Каким образом можно подключиться к базе данных *MySQL*?
- 4 Какое определение можно дать первичному ключу таблицы базы данных?

5 Какое определение можно дать внешнему ключу таблицы базы данных?

3 КУРСОВОЙ ПРОЕКТ

3.1 Назначение, цели и задачи курсового проекта

Разрабатываемое веб-приложение предназначено для реализации электронной коммерции (публикации) с базовым набором функций.

Цель выполнения курсового проекта – закрепление теоретических знаний и повышение практических навыков в разработке веб-приложений с использованием современных веб-технологий.

Задачи, решаемые студентом в процессе выполнения курсового проекта:

- изучение предметной области;
- обоснование выбора программного обеспечения;
- разработка структуры веб-сайта;
- разработка макета дизайна веб-сайта;
- верстка веб-сайта;
- реализация базы данных веб-сайта;
- реализация веб-сайта;
- продвижение веб-сайта;
- тестирование веб-сайта;
- выбор хостинга;
- оформление документации.

3.2 Требования к курсовому проекту

3.2.1 Требования к функциональным характеристикам

- 1 Регистрация и авторизация пользователя;
- 2 просмотр и редактирование товаров (публикаций);
- 3 просмотр и редактирование категорий товаров (публикаций);
- 4 поиск товаров (публикаций);
- 5 корзина пользователя;
- 6 оформление заказа;
- 7 история бронирования;
- 8 реализация счетчика посещения веб-сайта;
- 9 формирование статистической информации;
- 10 администрирование веб-сайта;
- 11 прогнозирование объемов продаж товара (посетителей веб-сайта).

3.2.2 Требования к эксплуатационным характеристикам

- Кроссплатформенность;
- модульность;
- расширяемость.

3.2.3 Требования к программному обеспечению

- 1 Фреймворк Laravel, Angular, Django, Spring Boot 2.
- 2 Сервер баз данных MySQL, PostgreSQL, SQLite.
- 3 Фреймворк Bootstrap.

3.3 Варианты заданий

3.3.1 Задание 1

- 1 Фирма, осуществляющая продажу товаров для физической культуры, спорта и туризма.
- 2 Фирма, осуществляющая продажу теле-, радиотоваров.
- 3 Фирма, осуществляющая продажу художественных товаров.
- 4 Фирма, осуществляющая продажу фото-, кинотоваров.
- 5 Фирма, осуществляющая продажу хозяйственных товаров.
- 6 Фирма, осуществляющая продажу вело-, мототехники.
- 7 Фирма, осуществляющая продажу кабельного оборудования.
- 8 Фирма, осуществляющая продажу спутникового оборудования.
- 9 Фирма, осуществляющая продажу электротоваров.
- 10 Фирма, осуществляющая продажу сельскохозяйственной техники.
- 11 Фирма, осуществляющая продажу канцелярских принадлежностей.
- 12 Фирма, осуществляющая продажу средств связи.
- 13 Фирма, осуществляющая продажу легковых автомобилей.
- 14 Фирма, предоставляющая риэлтерские услуги.
- 15 Фирма, предоставляющая туристические услуги.
- 16 Фирма, осуществляющая продажу строительных материалов.
- 17 Фирма, предоставляющая юридические услуги.
- 18 Фирма, осуществляющая продажу торгового оборудования.
- 19 Фирма, осуществляющая продажу книгопечатной продукции.
- 20 Фирма, осуществляющая продажу лакокрасочной продукции.
- 21 Фирма, осуществляющая продажу продуктов питания.
- 22 Фирма, осуществляющая продажу вычислительной техники.

- 23 Фирма, осуществляющая продажу теле-, видеоаппаратуры.
- 24 Фирма, осуществляющая продажу косметики.
- 25 Фирма, осуществляющая продажу музыкальных инструментов.

3.3.2 Задание 2

- 1 Интернет-магазин по продаже музыкальных инструментов.
- 2 Интернет-магазин по продаже вело-, мототехники.
- 3 Интернет-магазин по продаже книгопечатной продукции.
- 4 Интернет-магазин по продаже лакокрасочной продукции.
- 5 Интернет-магазин по продаже кабельного оборудования.
- 6 Интернет-магазин по продаже торгового оборудования.
- 7 Интернет-магазин по продаже средств связи.
- 8 Интернет-магазин по продаже вычислительной техники.
- 9 Интернет-магазин по продаже теле-, видеоаппаратуры.
- 10 Интернет-магазин по продаже косметики.
- 11 Интернет-магазин по продаже спутникового оборудования.
- 12 Интернет-магазин по продаже сельскохозяйственной техники.
- 13 Интернет-магазин по продаже канцелярских принадлежностей.
- 14 Интернет-магазин по продаже легковых автомобилей.
- 15 Интернет-магазин по продаже продуктов питания.
- 16 Интернет-магазин по продаже электротоваров.
- 17 Интернет-магазин по продаже строительных материалов.
- 18 Интернет-магазин по продаже теле-, радиотоваров.
- 19 Интернет-магазин по продаже художественных товаров.
- 20 Интернет-магазин по продаже фото-, кинотоваров.
- 21 Интернет-магазин по продаже хозяйственных товаров.
- 22 Интернет-магазин по продаже бытовой техники.
- 23 Интернет-магазин по продаже продуктов питания.
- 24 Интернет-магазин по продаже товаров для физической культуры, спорта и туризма.
- 25 Интернет-магазин по продаже сетевого оборудования.

3.3.3 Задание 3

- 1 Веб-сайт электронной публикации статей журнала «Техника молодёжи».
- 2 Веб-сайт электронной публикации статей детского журнала «Электроник».

- 3 Веб-сайт электронной публикации статей научного журнала «Наука и жизнь».
- 4 Веб-сайт электронной публикации статей спортивного журнала «Спорт».
- 5 Веб-сайт электронной публикации статей общества любителей путешествий «Вокруг света».
- 6 Веб-сайт электронной публикации статей общества любителей природы «Мир природы».
- 7 Веб-сайт электронной публикации научных статей университета «МГТУ имени Н. Э. Баумана».
- 8 Веб-сайт электронной публикации статей журнала «Здоровье».
- 9 Веб-сайт электронной публикации статей журнала «Футбол».
- 10 Веб-сайт электронной публикации статей журнала «Охота. Рыбалка».
- 11 Веб-сайт электронной публикации статей журнала «Финансовый бизнес».
- 12 Веб-сайт электронной публикации статей журнала «Автомир».
- 13 Веб-сайт электронной публикации статей журнала «Военно-стратегический анализ».
- 14 Веб-сайт электронной публикации статей журнала «Великая Отечественная».
- 15 Веб-сайт электронной публикации статей журнала «Computer World».
- 16 Веб-сайт электронной публикации статей журнала «Добрые советы».
- 17 Веб-сайт электронной публикации статей журнала «Наша кухня».
- 18 Веб-сайт электронной публикации статей журнала «Смена».
- 19 Веб-сайт электронной публикации статей журнала «Делаем сами».
- 20 Веб-сайт электронной публикации статей журнала «Аэрокосмическое обозрение».
- 21 Веб-сайт электронной публикации статей журнала «Военное дело».
- 22 Веб-сайт электронной публикации статей журнала «Наш сад».
- 23 Веб-сайт электронной публикации статей журнала «Техника молодёжи».
- 24 Веб-сайт электронной публикации статей журнала «Культурное обозрение».
- 25 Веб-сайт электронной публикации статей журнала «Садоводство».

3.4 Контрольные вопросы

- 1 Что понимается под сервисом в *Angular*?
- 2 Какие задачи выполняют сервисы в *Angular*?
- 3 Каким образом создается сервис в *Angular*?

- 4 Какое определение можно дать роутингу в *Angular*?
- 5 Каким образом производится настройка роутинга в *Angular*?

ЗАКЛЮЧЕНИЕ

Методические указания содержат теоретическое обоснование, варианты заданий для выполнения лабораторных работ и курсовых проектов, контрольные вопросы по дисциплине «Технологии разработки web-приложений».

Методические указания разработаны в соответствии с требованиями государственного образовательного стандарта по подготовке студентов по направлениям 09.03.04 «Программная инженерия», 09.03.03 «Прикладная информатика».

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- 1 Дронов В. А. Laravel 9. Быстрая разработка веб-сайтов на PHP / В. А. Дронов. – Санкт-Петербург : БХВ-Петербург, 2023. – 768 с.
- 2 Стаффер Мэтт. Laravel. Полное руководство / Мэтт Стаффер ; пер. с англ. – 3-е изд. – Санкт-Петербург : Питер, 2024. – 544 с.
- 3 Кириченко А. В. Laravel для web-разработчиков. Практическое руководство по созданию профессиональных сайтов / А. В. Кириченко, В. В. Дубовик. – Санкт-Петербург : Наука и Техника, 2021. – 432 с.
- 4 Дронов В. А. Laravel. Быстрая разработка современных динамических web-сайтов на PHP, MySQL, HTML и CSS / В. А. Дронов. – Санкт-Петербург : БХВ-Петербург, 2018. – 768 с.
- 5 Пономарев С. Н. Angular на примерах. Создаем web-приложение с нуля / С. Н. Пономарев. – Санкт-Петербург : Наука и техника, 2019. – 288 с.
- 6 Фримен А. Angular для профессионалов / А. Фримен. – Санкт-Петербург: Питер, 2018. – 800 с.
- 7 Чёрный Б. Профессиональный TypeScript. Разработка масштабируемых JavaScript-приложений / Б. Чёрный. – Санкт-Петербург : Питер, 2024. – 352 с.
- 8 Моисеев А. Angular и TypeScript. Сайтостроение для профессионалов / А. Моисеев, Я. Файн. – Санкт-Петербург : Питер, 2018. – 464 с.
- 9 Фримен А. Основы TypeScript / А. Фримен. – Санкт-Петербург : Прогресс книга, 2024. – 576 с.
- 10 Вандеркам Д. Эффективный TypeScript / Д. Вандеркам. – Санкт-Петербург : Питер, 2022. – 288 с.
- 11 Дронов В. А. Angular4. Быстрая разработка сверхдинамических web-сайтов на TypeScript и PHP / В. А. Дронов. – Санкт-Петербург : БХВ-Петербург, 2018. – 448 с.

12 Список привычек для трекера привычек (100+ полезных). – URL : <https://proaffirmacii.ru/voprosi-i-otveti/spisok-privyчек-dlya-trekera-privyчек-100-poleznyh> (дата обращения : 20.12.2024 г.).

Семахин Андрей Михайлович

ТЕХНОЛОГИИ РАЗРАБОТКИ WEB-ПРИЛОЖЕНИЙ

Методические указания
к выполнению лабораторных работ и курсовых проектов
для студентов направлений 09.03.03, 09.03.04

Редактор О. Г. Алексева

Подписано в печать	Формат 60x84 1/16	Бумага /80 г/м ²
Печать цифровая	Усл. печ. л. 2,37	Уч.-изд. л. 2,37
Заказ	Тираж 25	

Библиотечно-издательский центр КГУ.
640002, г. Курган, ул. Советская, 63/4.
Курганский государственный университет.