

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
Факультет прикладной математики и информатики  
Кафедра технологий программирования

Лабораторная работа № 7  
По курсу “Проектирование человеко-машинных интерфейсов”

**Проектирование и разработка веб-приложения и мобильного приложения: базовый функционал (арі, логика и интерфейс)**

Методические указания по выполнению лабораторной работы

Подготовила: Давидовская М.И.,  
Старший преподаватель кафедры ТП

Минск, 2022 г.

## Содержание

Введение.....	2
Рекомендации по выполнению лабораторной работы.....	2
План выполнения лабораторной работы:.....	3
Рекомендации по использованию инструментов разработки.....	4
Непрерывная интеграция, настройка, юнит-тесты и другие виды тестов.....	4
Запросы и данные.....	4
Препроцессоры.....	5
Краткая инструкция по экспорту в CSS.....	5
Frontend frameworks.....	5
Автоматизация frontend-разработки (taskrunners).....	6
Сервисы временной почты.....	6

## Введение

В результате выполнения лабораторных работ студент должен приобрести следующие навыки и умения:

- проектирование ПО на основе человеко-центрированного подхода;
- разработка прототипов интерфейсов,
- применение принципов и шаблоны проектирования взаимодействия;
- анализ и оценка пользовательского интерфейса ПО;
- проектирование и разработка мобильного приложения (HTML5, CSS3) и веб-приложения (PHP/Python (Django)/Ruby (Ruby on Rails)/Java(Java Spring Framework: Spark, Spring MVC)/C#(Asp.Net MVC)); css-фреймворки (Bootstrap, Zurb Foundation), js- фреймворки (jQuery, AngularJS, React, Knockout, Vue) и др.

Лабораторные работы связаны между собой единой темой из предложенного преподавателем списка или выбранной студентом самостоятельно при условии предварительного согласования с преподавателем. Входными данными для каждой последующей работы являются результаты предыдущей.

## Рекомендации по выполнению лабораторной работы

**Цель работы** — получить навыки проектирования и разработки логики мобильного и веб-приложения.

Выполнение лабораторной работы № 7 состоит из следующих этапов:

1. Начать реализацию проекта согласно техническому заданию, распределив задачи между участниками команды на разработку фронтенда и бекенда.
2. Реализовать запросы API для некоторых типов запросов, например, получение товара, списка товаров, сохранение товара, изменение товара.

3. Реализовать базовый функционал, обеспечивающий вывод данных (карточки, списки), фильтрацию, основные страницы, связанные с отображением основных сущностей, например, корзины, заказа, товара и т.д. Примерный перечень работ: добавление навигации, рендеринг данных, обработка форм, создание модели, вывод списка товаров, подключение клиентских скриптов, редактирование товара, модель корзины, вывод данных в корзине, обработка асинхронных скриптов, динамическое изменение корзины. Пример реализации веб-приложения – <https://habr.com/ru/post/444446/>
4. Формировать автоматическую сборку при публикации изменений в репозиторий, используя сервис непрерывной интеграции (Github Actions или иной бесплатный вариант).
5. Перед сборкой выполнять проверку качества кода с помощью сервиса Better Code Hub или другого.
6. Разработать автоматические тесты и формировать сборку только после их прохождения.
7. Создание отчёта, описывающего работы по всем пунктам данного задания и вклад каждого участника проекта, включая информацию по комитам из git-репозитория и распределение работ в рамках проекта

Для публикации на внешнем хостинге можно использовать бесплатные облачные платформы [OpenShift](#), [Heroku](#) или Google Cloud. Для развертывания окружения разработки локально использовать контейнеры на примере Docker или LXC/LXD. Продемонстрировать подключение к окружению разработки и запуск приложения на облачном сервисе.

## План выполнения лабораторной работы:

№	Задача
1.	Техническое задание
2.	Распределение задач на реализацию базового функционала и назначение ответственных
3.	Реализация задач на разработку фронтенда и бекенда, включая api, для вывода данных для одной сущности (например товара, продукта, счета и т.д.) и списков, включая каталог товаров, корзину и т.д.
4.	Дополнять итоговый отчет описанием выполненных работ

# Рекомендации по использованию инструментов разработки

Что должен уметь backend-разработчик

<https://habrahabr.ru/company/netologyru/blog/328426/>

[https://geekbrains.ru/posts/profession\\_web\\_developer](https://geekbrains.ru/posts/profession_web_developer)

## Непрерывная интеграция, настройка, юнит-тесты и другие виды тестов

[How to build a CI/CD pipeline with GitHub Actions in four simple steps](#)

[Непрерывная интеграция с GitHub Actions](#)

About continuous integration - <https://docs.github.com/en/actions/automating-builds-and-tests/about-continuous-integration>

[Непрерывная интеграция и развертывание настольных приложений с GitHub Actions](#)

Автоматизация сборки

[https://ru.wikipedia.org/wiki/%D0%90%D0%B2%D1%82%D0%BE%D0%BC%D0%B0%D1%82%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F\\_%D1%81%D0%B1%D0%BE%D1%80%D0%BA%D0%B8](https://ru.wikipedia.org/wiki/%D0%90%D0%B2%D1%82%D0%BE%D0%BC%D0%B0%D1%82%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F_%D1%81%D0%B1%D0%BE%D1%80%D0%BA%D0%B8)

<http://habrahabr.ru/post/155201/>

[Continuous Testing в CI/CD: что это, зачем нужно и как работает](#)

[Что такое непрерывное тестирование и зачем оно нужно вашим разработчикам](#)

[Непрерывная интеграция и тестирование ПО](#)

## Запросы и данные

### Документирование API

Курс по документированию REST API - <https://github.com/docops-hq/learnapidoc-ru> и <https://starkovden.github.io>

Swagger (OpenAPI 3.0) - <https://habr.com/ru/post/541592/>

Книга Сергея Константинова об API - <https://twirl.github.io/The-API-Book/API.ru.html>

Елена Бенкен. Тестирование и документирование API-методов с помощью Postman - <https://youtu.be/OWPmRyqERgY>

### Генератор данных

8 отличных сервисов для генерации данных

[https://quality-lab.ru/blog/8\\_services\\_for\\_data\\_generation/](https://quality-lab.ru/blog/8_services_for_data_generation/)

Генератор личности <https://randus.org>

Генератор личности <https://randomdatatools.ru>

Online test data generator <https://www.onlinedatagenerator.com>

[Сервис генерации тестовых данных](#)

SQL Test Data Generator Tool - <https://www.coderstool.com/sql-test-data-generator>

Десять классных генераторов тестовых данных <https://testengineer.ru/klassnye-generatory-testovyh-dannyh/>

## Postman

Клиент для тестирования запросов к сайтам - <https://www.getpostman.com/apps>

Кроме десктопных приложений есть расширение для браузера Chrome. Для Firefox можно использовать другие расширения, например HttPrequest

Использование postman для тестирования запросов -

<https://mindbox.fogbugz.com/default.asp?W1299>

## Препроцессоры

<https://htmlacademy.ru/courses/85>

[http://paulradzkov.com/2017/local\\_variables/](http://paulradzkov.com/2017/local_variables/)

<https://habrahabr.ru/post/214143/>

## Краткая инструкция по экспорту в css

1. Создайте файл style.less, в котором вы импортируете необходимые бутстрап-файлы (пусть они лежат в отдельной папке, чтобы не путаться). Например:

```
@import "bootstrap/less/variables.less"
@import "bootstrap/less/mixins.less"
@import "bootstrap/less/normalize.less"
@import "bootstrap/less/mixins.less"
@import "bootstrap/less/buttons.less"
```

Список всех файлов вы можете увидеть в [bootstrap.less](#). Если вам нужен весь бутстрап, вы можете просто этот bootstrap.less импортировать.

2. Пишите свои стили после импортов или в отдельном файле, и включите его в style.less с помощью импорта:

```
// Bootstrap
@import "bootstrap/less/variables.less"
...
@import "bootstrap/less/buttons.less"

// Project Styles
@import "project.less"
```

3. Скомпилируйте LESS в CSS с помощью, как уже посоветовали, [npm-пакета less](#) или утилиты [Prepros](#). Подключайте итоговый CSS-файл к вашей странице.

```
npm install -g less
lessc styles.less > styles.css
```

## Frontend frameworks

Как выбрать фреймворк для frontend-разработки <https://habrahabr.ru/post/277547/>

Топ 10 JavaScript-фреймворков и библиотек в 2022 <https://tproger.ru/articles/top-10->

[javascript-frejmworkov-i-bibliotek-v-2022/](#)

Популярные фреймворки JavaScript в 2020 <https://vc.ru/dev/147263-populyarnye-freymvorki-javascript>

Обзор 5 самых популярных JavaScript фреймворков и библиотек 2017  
<https://habrahabr.ru/post/321844/>

5 самых популярных фреймворков для JavaScript  
[https://geekbrains.ru/posts/5\\_js\\_frameworks](https://geekbrains.ru/posts/5_js_frameworks)

VUE — <https://habrahabr.ru/post/329452/>

## **Автоматизация frontend-разработки (taskrunners)**

Автоматизация для фронтендеров <https://www.youtube.com/watch?v=rJr0-e0ZJiU>  
<https://habrahabr.ru/post/251807/>

Level Up для новичков: gulp и requirejs <https://habrahabr.ru/post/264869/>  
<https://habrahabr.ru/company/2gis/blog/269743/>

Пособие по webpack <https://habrahabr.ru/post/309306/>

React с нуля. Настройка связи gulp + webpack + babel + react  
<https://medium.com/@artvaleyev/react-s-nulya-nastrojka-svyazki-gulp-webpack-babel-react-7be0203614bc>

<https://loftblog.ru/lessons/avtomatizaciya/>

## **Сервисы временной почты**

<https://tempmail.plus>

<https://temp-mail.org/>

<https://temp-mail.ru/>