

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 Мобильные и сетевые технологии

ОТЧЕТ

по лабораторной работе №1

по теме:

ВЁРСТКА САЙТА СРЕДСТВАМИ HTML, CSS И BOOTSTRAP

Вариант: Платформа для фитнес-тренировок и здоровья

Студент:

В.С. Корчагин

Преподаватель:

Должность

Д.И. Добряков

Санкт-Петербург 2025

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
1 Постановка задачи и выбранный вариант	4
1.1 Формулировка задачи	4
1.2 Описание выбранного варианта	4
2 Проектирование структуры сайта	6
2.1 Структура страниц и навигация	6
2.2 Выбор стека технологий	6
3 Реализация пользовательского интерфейса	8
3.1 Главная страница (index.html)	8
3.2 Страница поиска тренировок (workouts.html)	8
3.3 Страница тренировки (workout.html)	9
3.4 Страницы тренировочных планов (training-plans.html, training- plan.html)	9
3.5 Личный кабинет пользователя (dashboard.html)	10
3.6 Раздел блога (blog.html, blog-post.html)	10
3.7 Формы аутентификации (login.html, register.html)	11
4 Использование css и bootstrap для адаптивной вёрстки	13
4.1 Цветовая схема и типографика	13
4.2 Сетка и адаптивность	13
4.3 Компоненты Bootstrap и иконки	14
5 Использование javascript в рамках лабораторной работы 1	15
5.1 Инициализация страниц и переключение навигации	15
5.2 Динамическое наполнение карточек тренировок и блога	15
5.3 Фильтрация тренировок на стороне клиента	16
5.4 Подготовка к работе с личным кабинетом	16
ЗАКЛЮЧЕНИЕ	17

ВВЕДЕНИЕ

Современные веб-платформы для фитнеса и здоровья совмещают в себе несколько типов функциональности: личный кабинет пользователя, инструменты для подбора тренировок по параметрам, просмотр детальной информации по каждой тренировке и ведение блога о здоровье и питании. Пользователь ожидает, что интерфейс будет адаптивным, удобным и единообразным на разных устройствах.

Цель лабораторной работы №1 — разработать адаптивную вёрстку многостраничного сайта «Grizzly Gym» средствами HTML, CSS и Bootstrap для варианта «Платформа для фитнес-тренировок и здоровья». В рамках первой лабораторной работы требуется реализовать именно клиентскую часть (структура страниц и визуальное оформление), а также использовать JavaScript для организации базовых интерактивных элементов (навигация, фильтры, подготовка к работе с модальными окнами и формами).

В процессе выполнения работы были спроектированы и свёрстаны основные страницы платформы: главная страница, поиск тренировок с фильтрацией, детальная страница тренировки, блог о здоровье и питании, формы регистрации и входа, а также простой личный кабинет пользователя (dashboard). Вёрстка реализована на основе Bootstrap 5 и собственных CSS-стилей, обеспечивающих единый визуальный стиль бренда «Grizzly Gym».

1 Постановка задачи и выбранный вариант

1.1 Формулировка задачи

Согласно условию лабораторной работы №1 необходимо:

- выбрать один вариант предметной области из предложенного списка;
- разработать вёрстку сайта средствами HTML, CSS и Bootstrap;
- продумать и реализовать интерфейс, включающий:
 - вход и регистрацию пользователей;
 - личный кабинет пользователя с отображением прогресса и планов тренировок;
 - поиск тренировок с фильтрацией по уровню, типу и продолжительности;
 - страницу отдельной тренировки с видео, описанием и инструкциями;
 - блог о здоровье и питании.

Дополнительно разрешается использовать JavaScript для реализации необходимых элементов интерактивности (обработка форм, переключение элементов интерфейса, работа с фильтрами, открытие модальных окон и т.п.).

1.2 Описание выбранного варианта

В работе выбран вариант «Платформа для фитнес-тренировок и здоровья». Пользовательская часть системы должна поддерживать следующие сценарии:

- новый пользователь может зарегистрироваться и создать учётную запись;
- существующий пользователь может выполнить вход в систему;
- авторизованный пользователь получает доступ к личному кабинету, где отображается его прогресс и привязанные тренировочные планы;
- любой пользователь может перейти на страницу списка тренировок и подобрать тренировку по уровню подготовки, типу нагрузки (кардио, силовые и т.п.) и продолжительности;

- со страницы списка тренировок пользователь может открыть детальную страницу конкретной тренировки;
- пользователь может просматривать записи блога о здоровье, питании и восстановлении.

Все эти сценарии ориентированы на визуальную реализацию интерфейса в рамках первой лабораторной работы. Реальное взаимодействие с внешними API и сервером предполагается во второй лабораторной работе и в данном отчёте подробно не рассматривается.

2 Проектирование структуры сайта

2.1 Структура страниц и навигация

В основе веб-приложения лежит набор HTML-страниц, связанных единой навигационной панелью (navbar) и общим футером:

- `index.html` — главная страница с кратким описанием сервиса и быстрыми переходами к ключевым разделам;
- `workouts.html` — список всех тренировок с фильтрами по уровню, типу и продолжительности;
- `workout.html` — детальная страница отдельной тренировки;
- `training-plans.html` — список комплексных тренировочных планов;
- `training-plan.html` — детальная страница отдельного тренировочного плана;
- `blog.html` — список записей блога о здоровье и питании;
- `blog-post.html` — страница отдельной записи блога с комментариями;
- `login.html` и `register.html` — формы входа и регистрации;
- `dashboard.html` — личный кабинет пользователя.

Во всех страницах используется единый верхний навбар `navbar-custom` на Bootstrap с логотипом «Grizzly Gym» и ссылками на основные разделы: Workouts, Training plans, Blog, Dashboard, Login, Register. Для ряда ссылок применяются дата-атрибуты (`data-nav="login"`, `data-nav="dashboard"` и т.п.), позволяющие JavaScript-коду показывать и скрывать элементы в зависимости от состояния авторизации.

2.2 Выбор стека технологий

Для реализации пользовательского интерфейса были использованы следующие технологии:

- HTML5 — для описания структуры страниц, форм, карточек и компонентов интерфейса;
- CSS3 — для настройки цветовой схемы, типографики, размеров и отступов;

- Bootstrap 5.3 — для реализации адаптивной сетки, готовых компонентов (navbar, cards, buttons, grid) и базовой стилизации;
- Bootstrap Icons — для добавления пиктограмм (иконки активности, графиков, видео и т.п.) в интерфейс;
- JavaScript/TypeScript — для управления навбаром, фильтрами, заполнения карточек тренировок и блога и первичной логики личного кабинета.

Собственные стили оформлены в файле `css/main.css` и дополняют Bootstrap, задавая фирменную цветовую палитру «Grizzly Gym» и адаптивное поведение отдельных блоков (герой-блок, сетки карточек, дашборд).

3 Реализация пользовательского интерфейса

3.1 Главная страница (index.html)

Главная страница содержит крупный герой-блок `header.hero` с лозунгом «Train like a grizzly» и описанием платформы. Слева располагается текстовое описание, справа — логотип в виде медведя. Кнопки-призывы к действию оформлены в виде двух кнопок Bootstrap:

- «Start now» — ссылка на страницу регистрации;
- «Log in» — ссылка на страницу входа.

Ниже герой-блока расположены два раздела:

- блок «Why Grizzly Gym» с четырьмя карточками преимуществ (умные планы тренировок, отслеживание прогресса, видео-тренировки, здоровье и питание);
- блок «Featured Workouts» и «From the blog» — превью карточек тренировок и записей блога, которые динамически заполняются при помощи JavaScript.

Вёрстка главной страницы основана на сетке Bootstrap (`container`, `row`, `col-lg-*`) и собственных классах (`hero__content`, `hero__badge`, `feature-card`, `preview-grid`).

3.2 Страница поиска тренировок (workouts.html)

Страница `workouts.html` реализует основной сценарий поиска тренировок. Верхняя часть страницы содержит текстовый заголовок и кнопку перехода в личный кабинет. Ниже размещён фильтр из трёх выпадающих списков:

- фильтр по уровню (`filterLevel`): `beginner`, `intermediate`, `advanced`;
- фильтр по типу нагрузки (`filterType`): `cardio`, `strength`, `flexibility`, `mobility`;
- фильтр по продолжительности (`filterDuration`): «Under 20 min», «20–40 min», «Over 40 min».

Под блоком фильтров выводится сетка тренировок `div#workoutsGrid` с классом `workouts-grid`. Каждая тренировка представлена карточкой Bootstrap

с названием, уровнем, типом, продолжительностью и кратким описанием. Кнопка «View details» ведёт на детальную страницу конкретной тренировки.

3.3 Страница тренировки (workout.html)

Страница `workout.html` отвечает за отображение полной информации по конкретной тренировке. Разметка содержит:

- кнопку «Back to workouts» для возврата к списку тренировок;
- контейнер `div#workoutDetail`, в который JavaScript-код подставляет карточку с данными выбранной тренировки.

Карточка тренировки включает:

- заголовок с названием тренировки;
- строку с типом тренировки, уровнем и продолжительностью;
- описание тренировки;
- отдельный блок с текстовыми инструкциями по выполнению упражнений;
- при наличии ссылки — встроенный видеоплеер (через тег `iframe` в контейнере `ratio ratio-16x9 Bootstrap`).

Такое разделение позволяет пользователю быстро оценить сложность и длительность занятия, а также посмотреть видео-инструкцию.

3.4 Страницы тренировочных планов (training-plans.html, training-plan.html)

Страница `training-plans.html` отображает список комплексных тренировочных планов, каждый из которых объединяет несколько тренировок. Вёрстка использует:

- заголовок и краткое описание назначения страницы;
- информационную карточку с пояснением, как добавлять планы в личный кабинет;
- сетку `div#trainingPlansGrid`, где каждая карточка плана содержит название, краткое описание, информацию о количестве входящих тренировок и кнопки:

- «Add to my plans» — добавление плана в личный кабинет (функциональность готовится к использованию во второй лабораторной работе);
- «View details» — переход на страницу `training-plan.html`.

Детальная страница `training-plan.html` выводит заголовок плана, его описание, число включённых тренировок и список этих тренировок. Для каждой тренировки показывается название, тип нагрузки, уровень и продолжительность, а также ссылка на детальную страницу соответствующей тренировки.

3.5 Личный кабинет пользователя (`dashboard.html`)

Страница `dashboard.html` реализует упрощённый личный кабинет «Your den». Разметка состоит из двух основных колонок:

- боковая панель (`dashboard__sidebar`) с информацией о пользователе (инициалы, имя, e-mail) и быстрыми ссылками на поиск тренировок и блог, а также кнопкой «Logout»;
- основная область (`dashboard__content`), представляющая собой сетку карточек:
 - карточка «Progress tracking» — отображает последние значения веса, целевого веса, шагов и потреблённой воды, а также содержит форму для обновления этих данных;
 - карточка «Your training plans» — список добавленных пользователем тренировочных планов;
 - карточка с рекомендацией перейти к разделам тренировок и блога за вдохновением.

В рамках первой лабораторной работы акцент делается на вёрстке и логике отображения карточек. Подключение фактических данных пользователя во многом относится ко второй лабораторной работе.

3.6 Раздел блога (`blog.html`, `blog-post.html`)

Страница `blog.html` оформлена как сетка карточек с записями блога. Каждая карточка содержит:

- заголовок статьи;
- дату публикации;
- короткий текст-анонс (превью);
- кнопку «Read more» для перехода на полную версию записи.

Страница `blog-post.html` отображает содержимое одной статьи. Разметка включает:

- кнопку «Back to blog»;
- контейнер `div#blogPost` для заголовка, даты, автора и основного текста статьи;
- блок «Comments» с формой оставления комментария (`textarea` и кнопка отправки) и список уже оставленных комментариев.

Разметка комментариев выполнена в виде аккуратных блоков `comment-box` с именем автора, датой и текстом комментария.

3.7 Формы аутентификации (`login.html`, `register.html`)

Формы входа и регистрации (`login.html`, `register.html`) оформлены в едином стиле:

- обёртка `auth-wrapper` центрирует карточку формы по вертикали и горизонтали и задаёт фон с градиентной подложкой;
- карточка `auth-card` имеет скруглённые углы, тень и рамку, соответствующие общему стилю интерфейса;
- в верхней части карточки отображается логотип и текстовый подзаголовок («Welcome back» для входа и «Join the pack» для регистрации).

Форма регистрации включает поля:

- Name;
- Email;
- Password;
- Date of birth;
- Gender.

Форма входа содержит поля:

- Email;

– Password.

Под формами добавлены текстовые ссылки для перехода на противоположную форму («Already have an account? Login» / «Don't have an account? Register»), что улучшает сценарии навигации пользователей.

4 Использование css и bootstrap для адаптивной вёрстки

4.1 Цветовая схема и типографика

Основные настройки оформления определены в файле `css/main.css` через корневые CSS-переменные:

- `--color-brown` — основной акцентный коричневый цвет бренда;
- `--color-accent` — жёлтый акцент для кнопок и бейджей;
- `--color-dark` — тёмный фон навбара и героя;
- `--color-grey` — вспомогательный серый цвет для рамок и текста;
- `--color-light` — основной цвет фона страниц;
- `--font-heading` — шрифт заголовков (Montserrat/Roboto).

Тело документа оформлено шрифтом Roboto, для заголовков используется более плотное начертание и верхний регистр, что визуально поддерживает «брутальный» стиль Grizzly Gym. Отдельные элементы (бейджи, кнопки) оформлены через классы `.btn-brand-primary`, `.btn-brand-outline`, `.hero__badge`, `.card-tag` и др.

4.2 Сетка и адаптивность

Для сетки интерфейса используется комбинация Bootstrap Grid и современных возможностей CSS:

- компоненты Bootstrap (`container`, `row`, `col-*`) применяются для основных блоков (герой, карточки преимуществ, формы);
- пользовательские сетки реализованы через CSS Grid:
 - `.preview-grid` и `.workouts-grid` задают сетку карточек по схеме `repeat(auto-fit, minmax(...))`, что позволяет автоматически подстраивать количество колонок под ширину экрана;
 - `.blog-grid` оформляет сетку записей блога с адаптивным числом колонок;
 - `.dashboard__content` использует сетку для размещения нескольких карточек в одну, две или три колонки в зависимости от ширины экрана.

Для корректного отображения на мобильных устройствах используются медиазапросы:

- при ширине до 600px дашборд переключается на одноколонный режим (боковая панель уходит вверх, контент — ниже);
- при увеличении ширины (min-width: 768px, min-width: 992px) изменяется высота отступов, количество колонок и максимальная ширина боковой панели.

Таким образом, сайт корректно отображается как на смартфонах, так и на планшетах и десктопах.

4.3 Компоненты Bootstrap и иконки

Bootstrap используется не только для сетки, но и для готовых компонентов:

- `navbar navbar-expand-lg navbar-custom fixed-top` — фиксированная навигационная панель со сворачиванием в бургер-меню на малых экранах;
- `card` и `card-body` — карточки тренировок, планов и записей блога;
- `btn`, `btn-outline-dark`, `btn-sm` и собственные модификации — кнопки действий;
- `ratio ratio-16x9` — контейнер для встраивания видеоплеера на странице тренировки.

Библиотека Bootstrap Icons добавляет визуальные акценты: иконки активности (`bi-activity`), графика (`bi-graph-up`), видео (`bi-camera-video`), сердцебиения (`bi-heart-pulse`) и др. Они используются в карточках преимуществ на главной странице и помогают пользователю быстрее ориентироваться в типе представленной информации.

5 Использование javascript в рамках лабораторной работы 1

5.1 Инициализация страниц и переключение навигации

Логика инициализации страниц реализована в модуле `main.js` (сгенерированном из `src/main.ts`). После события `DOMContentLoaded` выполняются две основные функции:

- `setupNavbar()` — читает данные авторизации из локального хранилища и в зависимости от наличия токена показывает или скрывает пункты меню `Login`, `Register`, `Dashboard` и `Logout`, а также бейдж с именем пользователя;
- `routeInit()` — определяет текущую страницу по атрибуту `data-page` у тега `body` и вызывает соответствующую функцию инициализации (`initLanding`, `initWorkouts`, `initDashboard`, `initBlog` и т.д.).

Такой подход позволяет переиспользовать единый JavaScript-код на нескольких HTML-страницах и явно разделить логику по типу страницы.

5.2 Динамическое наполнение карточек тренировок и блога

Для главной страницы и страницы списка тренировок используются вспомогательные функции:

- `initLanding()` — загружает список тренировок и записей блога для блока превью и выводит только несколько первых элементов (до четырёх тренировок и трёх записей). При ошибке загрузки применяются запасные массивы с тестовыми данными, чтобы интерфейс оставался работоспособным;
- `renderWorkoutCards()` — рендерит карточки тренировок в переданный контейнер;
- `renderBlogCards()` — рендерит карточки записей блога.

В контексте первой лабораторной работы важно, что JavaScript отвечает за наполнение уже свёрстанных контейнеров `div#workoutsPreview`, `div#blogPreview`, `div#workoutsGrid` и др. данными в едином формате, а сама HTML-вёрстка карточек реализована средствами Bootstrap и собственных CSS-стилей.

5.3 Фильтрация тренировок на стороне клиента

На странице `workouts.html` фильтрация реализована полностью на стороне клиента. Функция `initWorkouts()`:

- загружает массив тренировок;
- сохраняет его во внутреннюю переменную;
- навешивает обработчики на три `select`-элемента (уровень, тип, продолжительность);
- при изменении значений применяет фильтрацию по выбранным условиям и перерисовывает карточки через `renderWorkoutCards()`.

Логика фильтрации по продолжительности учитывает три диапазона:

- «Under 20 min» — тренировки короче 20 минут;
- «20–40 min» — тренировки длительностью от 20 до 40 минут включительно;
- «Over 40 min» — тренировки длительностью более 40 минут.

Фильтрация реализована без перезагрузки страницы, что делает взаимодействие пользователя с интерфейсом более удобным и наглядным.

5.4 Подготовка к работе с личным кабинетом

Функция `initDashboard()` настраивает отображение личного кабинета:

- подставляет имя и e-mail пользователя в правой части боковой панели;
- вычисляет инициалы пользователя и выводит их в круглой плашке;
- заполняет карточку прогресса последними значениями (при их наличии) и выводит историю изменений;
- отображает список привязанных тренировочных планов.

С точки зрения первой лабораторной работы основной акцент делается на корректном отображении информации в заранее свёрстанной структуре дашборда, а также на согласованности верстки между статическими и динамически заполняемыми блоками. Реальные запросы к внешнему API и работа с сервером используются как задел для последующей лабораторной работы и в данном отчёте подробно не разбираются.

ЗАКЛЮЧЕНИЕ

В ходе выполнения лабораторной работы №1 была разработана адаптивная многостраничная вёрстка веб-платформы «Grizzly Gym» для варианта «Платформа для фитнес-тренировок и здоровья». Реализованы все ключевые пользовательские страницы: главная страница, поиск тренировок с фильтрацией, детальная страница тренировки, раздел тренировочных планов, блог о здоровье и питании, формы регистрации и входа, а также личный кабинет пользователя.

При помощи Bootstrap и собственных CSS-стилей сформирован единый визуальный стиль интерфейса: настроена цветовая палитра, типографика, карточки, кнопки и иконки. Использование CSS Grid и медиазапросов позволило обеспечить корректное отображение сайта на устройствах с разными размерами экрана.

JavaScript использован для базовой интерактивности: переключения элементов навигации в зависимости от состояния авторизации, динамического наполнения карточек тренировок и блога, а также фильтрации тренировок на стороне клиента. Вёрстка и клиентская логика построены таким образом, чтобы в дальнейших лабораторных работах можно было без изменений пользовательского интерфейса подключить реальные внешние API и серверную часть.

Таким образом, цели лабораторной работы №1, связанные с вёрсткой сайта средствами HTML, CSS и Bootstrap и реализацией ключевых страниц для выбранного варианта, достигнуты.