Приднестровский государственный университет им. Т.Г. Шевченко
Физико-технический институт

ОСНОВЫ ВЕБ-РАЗРАБОТКИ С ПРИМЕНЕНИЕМ ИИ-АССИСТЕНТОВ

Лабораторный практикум

Разработал:
ст. преподаватель
кафедры ИТ
Бричаг Д.В.

г. Тирасполь
2025 г.

Лабораторная работа №4

CSS-библиотеки: Bootstrap & Tailwind

Цель работы:

- Познакомиться с концепцией CSS-библиотек и фреймворков, понять их роль в ускорении фронтенд-разработки.
- Понять фундаментальную разницу между компонентным подходом (на примере Bootstrap) и utility-first подходом (на примере Tailwind CSS).
- Научиться быстро собирать пользовательские интерфейсы (UI) из готовых инструментов.

Теоретическая справка

Проблема сборки интерфейсов с нуля

Когда проект становится большим, написание всего CSS вручную приводит к проблемам:

- Повторение кода: Приходится снова и снова писать стили для одних и тех же элементов: кнопок, карточек, сеток.
- Отсутствие консистентности: Сложно поддерживать единый стиль для всех элементов на большом проекте.
- Медленная разработка: Создание адаптивной сетки, стилизация форм и других стандартных компонентов отнимает много времени.

Чтобы решить эти проблемы и ускорить разработку, были созданы CSS-библиотеки и фреймворки.

Что такое CSS-библиотека?

CSS-библиотека (или фреймворк) — это готовый набор CSS-классов (а иногда и JavaScript-компонентов), который можно подключить к проекту и сразу использовать для стилизации.

Вместо того чтобы писать

```
button {
    background-color: blue;
    color: white; padding: 10px;
}
```

вы просто добавляете HTML-элементу готовый класс, например,

```
<button class="btn btn-primary">
```

Разработчик не пишет стили с нуля, а конструирует интерфейс из заранее подготовленных «кирпичиков». Существует два основных подхода к созданию таких «кирпичиков».

Два подхода: Компонентный (Bootstrap) vs. Utility-First (Tailwind)

1. Компонентный подход (Bootstrap)

Представьте, что у вас есть коробка с готовыми деталями LEGO: целая дверь, окно, колесо. Вы просто берете их и ставите на свою модель.

Bootstrap работает так же. Он предоставляет готовые компоненты: .card, .navbar, .btn. Вы используете эти классы, и элемент сразу выглядит как полноценный, стилизованный компонент.

Плюсы:

- Очень быстро: можно собрать прототип или простой сайт за считанные минуты.
- Консистентность: все элементы выглядят единообразно.
- Низкий порог входа: легко начать, документация понятна.

Минусы:

- Сложность кастомизации: если вам нужна кнопка нестандартного вида, придется переопределять стили Bootstrap, что бывает неудобно.

- «Сайты на Bootstrap похожи друг на друга»: без глубокой кастомизации дизайн выглядит узнаваемо и шаблонно.
- Избыточность: Фреймворк подключает много стилей, которые вы можете даже не использовать.

2. Utility-First подход (Tailwind CSS)

А теперь представьте, что у вас коробка с самыми мелкими деталями LEGO: кубики 1x1 разных цветов. Вы можете построить из них что угодно, но на сборку окна уйдет больше времени, чем если бы вы взяли готовое.

Tailwind — это набор «утилитарных» классов, каждый из которых выполняет ровно одну маленькую задачу: p-4 (padding: 1rem), bg-red-500 (background-color: #ef4444), flex (display: flex), rounded-lg (border-radius: 0.5rem).

Вы не используете готовый компонент .card, а собираете его сами прямо в HTML:

```
<div class="bg-white p-6 rounded-lg shadow-md">...</div>
```

Плюсы:

- Полная свобода дизайна: Вы можете создать абсолютно любой уникальный интерфейс.
- Никакого переопределения стилей: Вы не боретесь с фреймворком, а строите из его примитивов.
- Оптимизация: В итоговый CSS-файл попадают только те классы, которые вы реально использовали.

Минусы:

- «Раздутый» HTML: Классов у одного элемента может быть очень много.
- Выше порог входа: Нужно выучить и привыкнуть к названиям утилитарных классов.

- Первоначальная настройка медленнее: Сборка простых компонентов занимает больше времени, чем в Bootstrap.

Характеристика	**Bootstrap**	**Tailwind CSS**
Подход	Компонентный («готовые блоки»)	Utility-First («микро-детали»)
HTML	Чистый, классы-компоненты (<button class="btn">)	«Замусоренный», много классов (<button class="py-2 px-4...">)
CSS	Готовый CSS-файл, который нужно кастомизировать	Стили пишутся прямо в HTML через классы
Скорость прототипирования	Очень высокая	Средняя
Гибкость дизайна	Низкая (без кастомизации)	Очень высокая
Лучше всего подходит для	Админ-панелей, внутренних систем, быстрых лендингов	Уникальных дизайн-проектов, долгосрочных продуктов

Таблица – Bootstrap vs Tailwind – краткое сравнение

Как это работает на практике

Для обеих библиотек самый простой способ начать — подключить их через CDN (Content Delivery Network). Не нужно ничего устанавливать или компилировать.

1. Создается файл index.html.
2. В тег <head> добавляется ссылка на CSS-файл библиотеки.
3. Для Bootstrap также добавляется ссылка на JS-файл перед закрывающим тегом </body> для работы интерактивных компонентов (меню, модальные окна).
4. Разработчик добавляет HTML-элементам нужные классы из документации.

Пример подключения Bootstrap:

```
<head>
  <link href="[https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/boot-
strap.min.css](https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css)"
rel="stylesheet">
</head>
<body>
  <!-- Ваш контент -->
  <script src="[https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bun-
dle.min.js](https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bun-
dle.min.js)"></script>
</body>
```

Пример подключения Tailwind CSS:

```
<head>
  <script src="[https://cdn.tailwindcss.com](https://cdn.tailwindcss.com)"></script>
</head>
<body>
  <!-- Ваш контент -->
</body>
```

Практическая часть

Сформируем техническое задание. Необходимо:

1. Создать папку проекта и два HTML-файла: bootstrap.html и tailwind.html.
2. Сверстать простой «лендинг продукта» на Bootstrap, используя его готовые компоненты (navbar, container, card, button).
3. Сверстать точную копию этого же лендинга на Tailwind CSS, подбирая утилитарные классы для воссоздания внешнего вида.
4. Сравнить ощущения от процесса разработки и описать выводы в отчёте.

Подготовка проекта

Создайте новую папку lab4

Внутри создайте две подпапки:

- /bootstrap-landing
- /tailwind-landing

Инициализируйте Git-репозиторий

1. Создание лендинга на Bootstrap

1. Используйте ИИ (например, ChatGPT, Copilot или Cursor) с промптом:

```
Сгенерируй адаптивный лендинг для вымышленного продукта (например, "AI Assistant Pro") с использованием Bootstrap 5.
Пусть будет хедер, секция с описанием, блок преимуществ (3 карточки), футер.
Добавь теги meta для адаптивности.
```

2. Убедитесь, что страница содержит минимальный набор компонентов:
   - Навигационную панель (<nav class="navbar">...).
   - Главный экран («hero section») с заголовком и кнопкой (..., <button class="btn btn-primary">).
   - Секцию с описанием продукта (<section class="about">…)
   - Секцию с тремя карточками преимуществ продукта (...).

- Подвал с навигацией (<footer>...).

3. Полученный код вставьте в index.html в папке /bootstrap-landing.
4. Измените текст, изображения и цвета под свой стиль.
5. Проверьте адаптивность (сжмите окно браузера, проверьте на телефоне).

Примерный код карточки на Bootstrap:

```
<div class="col-md-4">
  <div class="card shadow-sm">
    <div class="card-body">
      <h5 class="card-title">Преимущество 1</h5>
      <p class="card-text">Краткое и понятное описание ключевой особенности нашего заме-
чательного продукта.</p>
      <a href="#" class="btn btn-outline-primary">Узнать больше</a>
    </div>
  </div>
</div>
```

2. Создание аналога на Tailwind CSS

1. Установите Tailwind по официальной инструкции или используйте CDN.

Пример для CDN:

```
<script src="https://cdn.tailwindcss.com"></script>
```

2. Сгенерируйте аналогичный лендинг с ИИ:

```
Создай адаптивный лендинг продукта "AI Assistant Pro" с использованием Tailwind CSS.
Включи хедер, секцию описания, блок преимуществ (3 карточки), футер.
Используй utility-first классы и сделай всё адаптивным.
```

3. Вставьте код в index.html в /tailwind-landing и также адаптируйте вручную.

Примерный код той же карточки на Tailwind:

```
<div class="bg-white border border-gray-200 rounded-lg shadow-sm">
  <div class="p-4">
    <h5 class="text-xl font-semibold mb-2">Преимущество 1</h5>
    <p class="text-gray-700 mb-4">Краткое и понятное описание ключевой особенности
нашего замечательного продукта.</p>
    <a href="#" class="text-blue-600 border border-blue-600 hover:bg-blue-600
```

```
hover:text-white font-semibold py-2 px-4 rounded">
      Узнать больше
    </a>
  </div>
</div>
```

3. Использование ИИ для помощи развития страницы

Используйте AI-ассистентов для создания схожих блоков и добавления новых к имеющейся странице.

Примеры промптов:

```
«Сгенерируй блок отзывов для страницы, использующей библиотеку Bootstrap»
```

```
«Создай контактную форму для лендинга с Tailwind»
```

```
«Создай карточки цен продукта для лендинга»
```

```
«Сгенерируй блок портфолио для лендинга»
```

4. Работа с git и GitHub

1. Инициализируйте репозиторий в папке проекта:

```
git init
git add .
git commit -m "Lab4: initial commit with landing pages"
```

2. Создайте новый репозиторий на GitHub.
3. Свяжите локальный проект с GitHub и отправьте файлы:

```
git remote add origin
[https://github.com/](https://github.com/)<ваш_логин>/<имя_репозитория>.git
git branch -M main
git push -u origin main
```

Отчёт (в README.md)

В файле README.md вашего репозитория добавьте:

- Заголовок: Лабораторная работа №4: CSS-библиотеки.
- Скриншоты: Как выглядит ваш лендинг с разными библиотеками.
- Ответы на вопросы:
  - Какой подход (компонентный Bootstrap или utility-first Tailwind) показался вам удобнее для этой задачи и почему?
  - Приведите пример кода для одного и того же элемента (например, кнопки) на Bootstrap и на Tailwind из вашей работы. В чем ключевое различие в разметке?
  - С какими сложностями вы столкнулись при работе с каждой из библиотек?
  - Какой блок вы добавили дополнительно?
  - Какие запросы вы использовали для помощи ИИ? Насколько полезными были ответы?

Результаты работы

В итоге у вас должно быть:

1. Файл bootstrap.html со страницей, собранной на Bootstrap.
2. Файл tailwind.html с визуально идентичной страницей на Tailwind CSS.
3. Репозиторий на GitHub с обоими файлами проекта.
4. Файл README.md с отчётом, скриншотами и ответами на вопросы.

Критерии оценки

- Сверстан лендинг на Bootstrap (2 балла).
- Сверстан визуально идентичный аналог на Tailwind CSS (2 балла).
- В отчёте дан развернутый ответ-сравнение двух подходов (2 балла).
- Код выложен на GitHub (2 балла).
- README.md содержит отчёт, скриншоты и ответы на все вопросы (2 балла).