

# Лабораторная работа 1: Вёрстка сайта

---

**Студент:** Даньшин Семён

**Группа:** К3440

**Дата:** 12 января 2026 г.

## Тема работы

Вёрстка сайта средствами HTML, CSS и Bootstrap (в данном проекте используется современный аналог - Tailwind CSS + HeroUI).

## Вариант задания

### Вариант 6: Свой вариант (Мультибанкинговая платформа)

Приложение представляет собой агрегатор банковских счетов с возможностью управления финансами, просмотра истории транзакций и использования ИИ-ассистента.

## Реализованные страницы

В рамках лабораторной работы были сверстаны следующие ключевые страницы:

### 1. Страница авторизации и регистрации

- **Вход:** Форма с полями Client ID и Пароль. Возможность входа в демо-режим.
- **Регистрация:** Форма создания нового аккаунта.
- **Особенности:** Адаптивная карточка по центру экрана, валидация полей, переключение между режимами.

### Код компонента (**components/auth/LoginCard.tsx**):

```
<Card className="w-full">
  <CardHeader className="flex flex-col items-start">
    <p className="text-xl font-bold">Войти</p>
    <p className="text-small text-default-500">Используйте данные своего аккаунта</p>
  </CardHeader>
  <CardBody>
    <Form className="flex flex-col gap-4" onSubmit={handleSubmit}>
      <Input label="Client ID" isRequired />
      <Input label="Пароль" type="password" isRequired />
      <Button type="submit" color="primary" fullWidth>Войти</Button>
    </Form>
  </CardBody>
</Card>
```

### 2. Главная страница (Дашборд)

- **Верхняя навигация (Navbar):** Логотип, ссылки на разделы, переключатель темы, кнопка выхода.
- **Баланс:** Отображение общего баланса по всем счетам.
- **Банковские карты:** Список подключенных банков и карт в виде карточек.
- **Виджет синхронизации:** Статус подключения к банкам.

### 3. История транзакций

- **Таблица/Список:** Список операций с фильтрацией по дате и категории.
- **Детализация:** Отображение суммы, категории, даты и статуса операции.
- **Графики:** Визуализация расходов по категориям.

### 4. Ассистент

- **Чат-интерфейс:** Окно диалога с виртуальным ассистентом.
- **Боковая панель:** Доступна на десктопе справа, на мобильных - через нижнее меню.

## Использованные технологии

### Tailwind CSS

Вместо классического CSS или Bootstrap в проекте используется Tailwind CSS. Это utility-first фреймворк, который позволяет быстро верстать адаптивные интерфейсы прямо в HTML/JSX.

#### Пример использования классов:

```
<div className="relative flex flex-col h-full flex-grow overflow-auto
scrollbar-hide">
  <Navbar />
  <main className="container mx-auto max-w-7xl pt-4 px-4 sm:pt-6 sm:px-6
flex-grow">
    {children}
  </main>
</div>
```

- **flex flex-col** - Flexbox колонка
- **h-full** - высота 100%
- **container mx-auto** - центрированный контейнер
- **px-4 sm:px-6** - адаптивные отступы (padding)

### HeroUI

В качестве библиотеки компонентов используется HeroUI (ранее NextUI), которая построена на базе Tailwind CSS. Она предоставляет готовые, доступные и красивые компоненты:

- **Navbar**
- **Card, CardBody**
- **Button**
- **Input**

- **Modal**

## Адаптивность

Вёрстка полностью адаптивна и поддерживает:

- **Desktop:** Полноценная навигация в шапке, боковая панель ассистента.
- **Mobile:** Нижняя навигационная панель (**BottomNavigation**), ассистент открывается в Drawer (выдвижной панели).

### Реализация адаптивной навигации:

```
{/* Desktopная навигация в Navbar */}
<ul className="hidden sm:flex gap-4 justify-start ml-2">
  {siteConfig.navItems.map(...)}
</ul>

{/* Мобильная навигация в BottomNavigation */}
<div className="xl:hidden fixed flex bottom-1 w-full z-50 justify-center">
  <Tabs ... />
</div>
```

## JS-вставки

В соответствии с заданием, продуманы моменты, требующие JS:

1. **Открытие модальных окон** - через состояние React (**isOpen**).
2. **Переключение табов** - навигация в мобильном меню.
3. **Мобильное меню** - открытие/закрытие Drawer.
4. **Валидация форм** - проверка полей перед отправкой.

## Заключение

В ходе выполнения лабораторной работы была реализована адаптивная вёрстка сложного веб-приложения банковской тематики. Использование Tailwind CSS и компонентного подхода позволило создать гибкий и поддерживаемый интерфейс, соответствующий современным стандартам веб-разработки.