

Домашняя работа 1: CSS Flexbox, CSS Grid, Git

Студент: Даньшин Семён

Группа: K3440

Описание задания

В рамках данной домашней работы необходимо было:

1. Пройти игру Flexbox Froggy (<https://flexboxfroggy.com/#ru>)
2. Пройти игру CSS Grid Garden (<https://cssgridgarden.com/#ru>)
3. Пройти первые 4 урока + все уроки по удалённым репозиториям в Learn Git Branching (https://learngitbranching.js.org/?locale=ru_RU)

1. CSS Flexbox

Игра Flexbox Froggy

Все уровни игры Flexbox Froggy успешно пройдены. В процессе прохождения были изучены следующие свойства Flexbox:

- `justify-content` - выравнивание элементов по главной оси
- `align-items` - выравнивание элементов по поперечной оси
- `flex-direction` - направление главной оси
- `order` - порядок элементов
- `align-self` - индивидуальное выравнивание элемента
- `flex-wrap` - перенос элементов на новую строку
- `flex-flow` - комбинация `flex-direction` и `flex-wrap`
- `align-content` - выравнивание строк в контейнере

Примеры использования Flexbox в проекте BankingThing

Пример 1: Навигационная панель (navbar.tsx)

```
<NavbarContent className="basis-1/5 sm:basis-full" justify="start">
  <NavbarBrand as="li" className="gap-3 max-w-fit">
    <NextLink className="flex justify-start items-center gap-1" href="/">
      <InteractiveLogo/>
    </NextLink>
  </NavbarBrand>
  <ul className="hidden sm:flex gap-4 justify-start ml-2">
    {pathname !== "/auth" && siteConfig.navItems.filter(item => item.label
    !== "Ассистент").map((item) => (
      <NavbarItem key={item.href}>
        <NextLink
          className={clsx(
            linkStyles({ color: "foreground" }),
            "data-[active=true]:text-primary data-[active=true]:font-
```

```

medium"
    })
    color="foreground"
    href={item.href}
  >
    {item.label}
  </NextLink>
</NavbarItem>
  )})
</ul>
</NavbarContent>

```

Использованные Tailwind классы:

- **flex** - включает flexbox
- **justify-start** - выравнивание по началу главной оси
- **items-center** - выравнивание по центру поперечной оси
- **gap-3, gap-4, gap-1** - отступы между элементами

Пример 2: Нижняя навигация (BottomNavigation.tsx)

```

<div className="xl:hidden fixed flex bottom-1 w-full z-50 justify-center">
  <Tabs
    aria-label="Bottom Navigation"
    items={siteConfig.navMenuItems}
    selectedKey={pathname}
    size="sm"
    classNames={{
      tab: "h-12 px-2 min-w-12",
      tabList: "bg-content1/30 backdrop-blur-xs gap-1 p-1 rounded-2xl
shadow-lg border border-default-200",
      cursor: "w-full bg-primary/20 rounded-xl",
    }}
  >
    {(item) => {
      const Icon = iconMap[item.icon];
      const isAssistant = item.label === "Ассистент"
      const assistCol = isAssistant ? "text-secondary-600/90" : ""
      return (
        <Tab
          key={item.href}
          href={isAssistant ? undefined : item.href}
          onClick={isAssistant ? openFromBottom : undefined}
          title={
            <div className={`flex flex-col items-center ${assistCol}`>
              {Icon && <Icon className={`w-6 h-6`} />}
              <span className={`text-[10px]`}>{item.label}</span>
            </div>
          }
        >
      )
    }}
  </>
</div>

```

```
    }}  
  </Tabs>  
</div>
```

Использованные Tailwind классы:

- **flex** - включает flexbox
- **flex-col** - вертикальное направление (column)
- **items-center** - центрирование по поперечной оси
- **justify-center** - центрирование по главной оси

Пример 3: Layout приложения (layout.tsx)

```
<div className="relative flex flex-row h-full">  
  <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>  
    <footer className="w-full flex items-center justify-center pb-20 pt-10  
xl:pb-3">  
      <Link  
        isExternal  
        className="flex items-center gap-1 text-current"  
        href="https://github.com/semendanshin/MultiBankingThing"  
      >  
        <span className="text-default-600">Powered by</span>  
        <p className="text-primary">Domino Squad</p>  
      </Link>  
    </footer>  
  </div>  
  <AssistantSidebar />  
</div>
```

Использованные Tailwind классы:

- **flex flex-row** - горизонтальный flexbox
- **flex flex-col** - вертикальный flexbox
- **flex-grow** - элемент растягивается, занимая доступное пространство
- **items-center** - центрирование по поперечной оси
- **justify-center** - центрирование по главной оси

2. CSS Grid

Игра CSS Grid Garden

Все уровни игры CSS Grid Garden успешно пройдены. В процессе прохождения были изучены следующие свойства Grid:

- `grid-column-start` / `grid-column-end` - позиционирование по колонкам
- `grid-row-start` / `grid-row-end` - позиционирование по строкам
- `grid-column` / `grid-row` - сокращённая запись
- `grid-area` - комплексное позиционирование
- `order` - порядок элементов
- `grid-template-columns` / `grid-template-rows` - определение размеров колонок и строк
- `grid-template` - сокращённая запись шаблона

Примеры использования Grid в проекте BankingThing

В данном проекте используется преимущественно Flexbox для компоновки, так как UI построен на библиотеке Tailwind CSS и HeroUI, которые широко применяют Flexbox. Однако Grid используется через утилитные классы Tailwind для создания сеток:

Пример: Возможное использование Grid для карточек

В проекте можно встретить компоновку элементов, которая может быть реализована через Grid. Например, в компоненте ConsentRow используется выравнивание элементов, которое могло бы быть реализовано через Grid:

```
<div className="flex items-center justify-between w-full">
  <p className="font-bold">{title}</p>
  <Chip color={isAllowed ? "success" : "danger"} variant="flat">
    {isAllowed ? "предоставлен" : "не предоставлен"}
  </Chip>
</div>
```

Альтернативная реализация через Grid:

```
<div className="grid grid-cols-2 gap-4 w-full items-center">
  <p className="font-bold">{title}</p>
  <Chip color={isAllowed ? "success" : "danger"} variant="flat"
  className="justify-self-end">
    {isAllowed ? "предоставлен" : "не предоставлен"}
  </Chip>
</div>
```

Использованные Tailwind классы:

- `grid` - включает CSS Grid
- `grid-cols-2` - создаёт две колонки
- `gap-4` - отступы между элементами
- `justify-self-end` - выравнивание элемента по концу ячейки

3. Git

Learn Git Branching

Пройдены следующие уроки:

- Первые 4 урока основного курса (Introduction Sequence)
- Все уроки по удалённым репозиториям (Remote)

Изученные команды Git

Основные команды:

- `git commit` - создание коммита
- `git branch` - создание ветки
- `git checkout` / `git switch` - переключение между ветками
- `git merge` - слияние веток
- `git rebase` - перебазирование веток

Команды для работы с удалёнными репозиториями:

- `git clone` - клонирование репозитория
- `git fetch` - получение изменений с удалённого репозитория
- `git pull` - получение и слияние изменений
- `git push` - отправка изменений на удалённый репозиторий
- `git remote` - управление удалёнными репозиториями

Примеры использования Git в проекте BankingThing

Проект BankingThing активно использует Git для контроля версий. Структура проекта организована в монорепозиторий с несколькими сервисами:

```
BankingThing/  
├── bank-connector/  
├── core-service/  
├── llm-agent/  
├── syncer/  
└── ui/
```

Каждый сервис имеет свой Makefile и независимую структуру, но все управляются из корневого репозитория.

Пример коммитов в проекте:

Судя по структуре проекта и коду, используются осмысленные коммиты, разделяющие функциональность:

- Отдельные коммиты для frontend компонентов (ui/)
- Отдельные коммиты для backend сервисов (bank-connector, core-service)

- Коммиты для настройки инфраструктуры (compose.yaml, Dockerfile)

Заключение

В рамках выполнения домашней работы были изучены:

1. **CSS Flexbox** - мощный инструмент для создания гибких макетов. В проекте BankingThing активно используется через Tailwind CSS для создания адаптивной навигации, карточек и компоновки элементов.
2. **CSS Grid** - система компоновки для создания двумерных сеток. Хотя в проекте преобладает Flexbox, Grid также доступен через Tailwind утилиты.
3. **Git** - система контроля версий, необходимая для командной разработки. Проект BankingThing демонстрирует правильную организацию репозитория с чёткой структурой и разделением на сервисы.

Все три технологии являются фундаментальными для современной веб-разработки и активно применяются в реальных проектах.