

Технология создания

Дизайн и вёрстка производилась одновременно. В ходе разработки использовались следующие инструменты:

Visual Studio Code - редактор кода для кроссплатформенной разработки веб и облачных приложений. Включает в себя отладчик, инструменты для работы с Git, подсветку синтаксиса и средства для рефакторинга. Имеет широкие возможности для кастомизации: пользовательские темы, сочетания клавиш и файлы конфигурации.

Live Server – расширение для VSCode, при сохранении файла обновляет страницу в браузере.

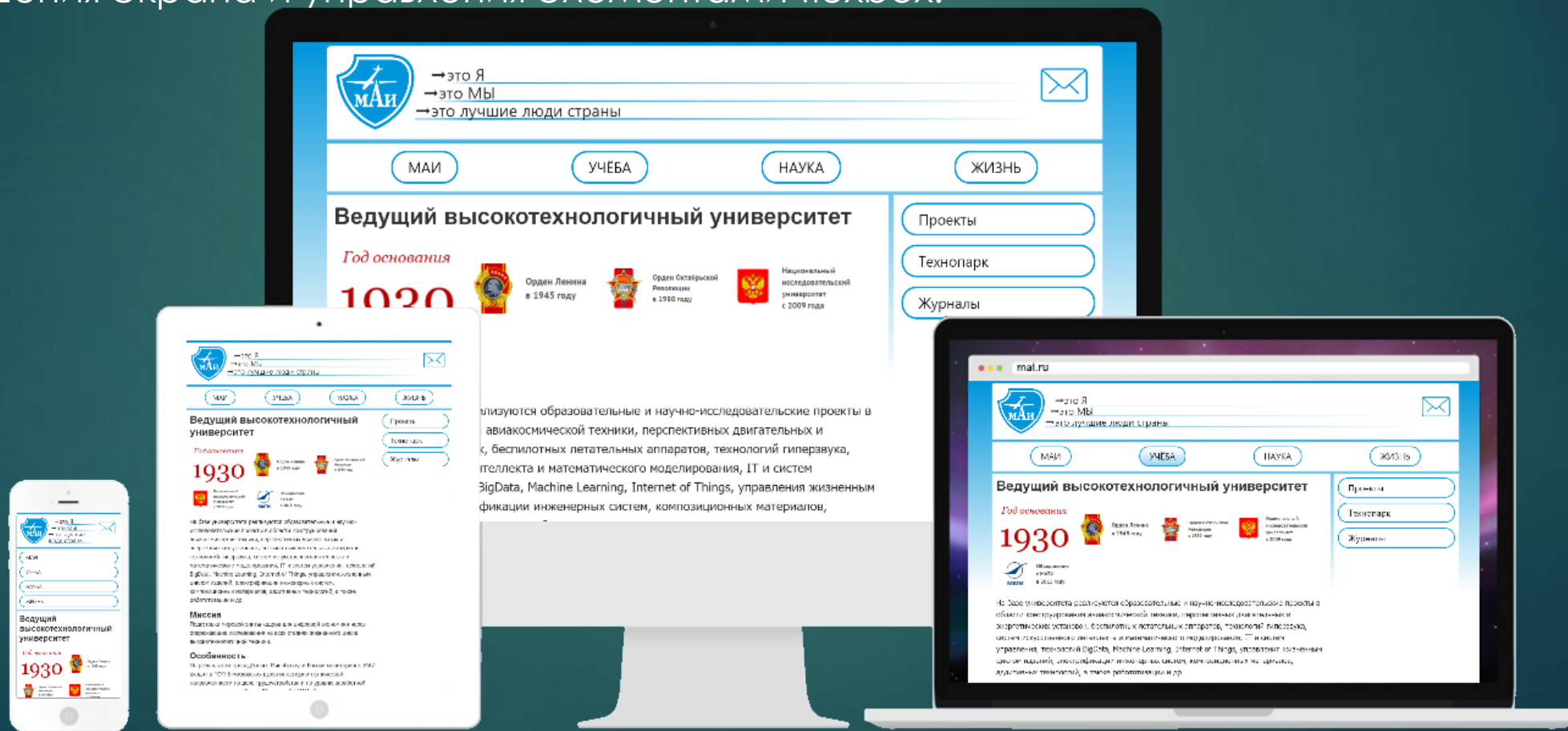
debug.css – подключается в файл html для подсветки границ элементов на стадии разработки.

reset.css – сбрасывает стили браузера по умолчанию, остаётся в конечном результате.

АДАПТИВНЫЙ ДИЗАЙН

Сайт корректно отображается на любых устройствах — стационарных компьютерах, ноутбуках, планшетах, мобильных телефонах.

Достигается за счёт автоматического изменения стилей при изменении разрешения экрана и управления элементами flexbox.



Исходный код index.html

```
<!doctype html>
<html lang="ru">
<head>
  <meta charset="urf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <meta name="theme-color" content="#0095DA">
  <meta name="description" content="Московский авиационный институт (национальный
исследовательский университет)">
  <title>МАИ</title>
  <link rel="stylesheet" href="style/reset.css" type="text/css">
  <link rel="stylesheet" href="style/style.css" type="text/css">
  <link rel="icon" href="favicon.ico">
</head>
<body>

<header>
  <div id="logo">
    <a href="/">
      
    </a>
  </div>
  <div id="slogan">
    <ul>
      <li>это Я</li>
      <li>это Мы</li>
      <li>это лучшие люди страны</li>
    </ul>
  </div>
  <div id="contact">
    <a href="mailto:mai@mai.ru">
      
    </a>
  </div>
</header>
```

```
<nav>
  <ul>
    <li class="button"><a href="main">МАИ</a></li>
    <li class="button"><a href="study">УЧЁБА</a></li>
    <li class="button"><a href="science">НАУКА</a></li>
    <li class="button"><a href="life">ЖИЗНЬ</a></li>
  </ul>
</nav>

<main>
  <article>
  </article>
  <aside>
    <ul>
      <li class="button"><a href="project">Проекты</a></li>
      <li class="button"><a href="tech">Технопарк</a></li>
      <li class="button"><a href="journal">Журналы</a></li>
    </ul>
  </aside>
</main>

<footer>
  © 2019 МАИ
</footer>
<script src="js/client.js"></script>
</body>
</html>
```

Использование градиентов

CSS-градиент представляет собой переходы от одного цвета к другому. Градиенты создаются с помощью функций `linear-gradient()` и `radial-gradient()`. Градиентный фон можно устанавливать в свойствах `background`, `background-image`, `border-image` и `list-style-image`.

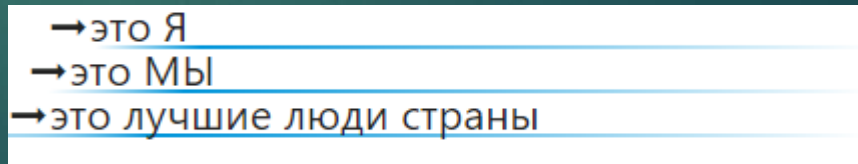
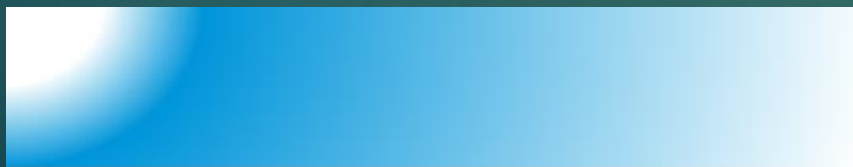
Линейный градиент используется в фоне страницы и навигационных кнопках.

```
background: linear-gradient(to bottom, #0095DA, white 70%);
```



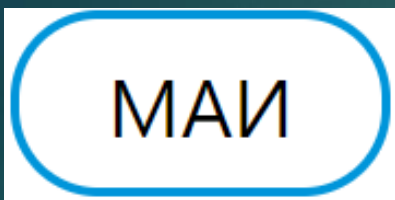
Радиальный градиент используется в шапке сайта, для создания декоративных линий.

```
background: radial-gradient(circle at left top, white 50px, #0095DA 100px, white)
```



Ролловер кнопки

Ролловер – (синоним английского термина Rollover – опрокидывание) – смена картинки на странице сайта при наведении/нажатии на нее курсора мыши. Вместо изображений использовался CSS градиент.



```
nav ul > li {  
    background: white;  
}
```

нормальное
состояние



```
nav ul > li:hover {  
    background: linear-gradient(to bottom, white 10%,  
    #0095DA 250%);  
}
```

при наведении



```
nav ul > li:active {  
    background: linear-gradient(to top, white 10%, #0095DA  
250%);  
}
```

при нажатии

Векторная графика

В макете использовалась векторная графика logo.svg, mail.svg.

Векторная графика — способ представления объектов и изображений (формат описания) в компьютерной графике, основанный на математическом описании элементарных геометрических объектов, обычно называемых примитивами, таких как: точки, линии, сплайны, кривые Безье, круги и окружности, многоугольники.

Преимущества:

- небольшой размер файла
- нет потери качества при увеличении
- Параметры объектов могут быть легко изменены



logo.svg



mail.svg

Flexbox

CSS flexbox (Flexible Box Layout Module) — модуль макета гибкого контейнера — представляет собой способ компоновки элементов, в основе лежит идея оси.

Flexbox состоит из гибкого контейнера (flex container) и гибких элементов (flex items). Гибкие элементы могут выстраиваться в строку или столбик, а оставшееся свободное пространство распределяется между ними различными способами.

Модуль flexbox позволяет решать следующие задачи:

- Располагать элементы в одном из четырех направлений: слева направо, справа налево, сверху вниз или снизу вверх.
- Переопределять порядок отображения элементов.
- Автоматически определять размеры элементов таким образом, чтобы они вписывались в доступное пространство.
- Решать проблему с горизонтальным и вертикальным центрированием.
- Переносить элементы внутри контейнера, не допуская его переполнения.
- Создавать колонки одинаковой высоты.
- Создавать прижатый к низу страницы подвал сайта.

Изменение CSS при разных разрешениях экрана

```
@media (max-width: 650px) {  
  header {  
    border-radius: 0;  
  }  
  nav ul {  
    flex-flow: column;  
    justify-content: space-between;  
    height: 200px;  
  }  
  main {  
    flex-flow: column;  
  }  
  article {  
    width: auto;  
    flex-grow: 1;  
  }  
  aside {  
    width: auto;  
    flex-grow: 0;  
  }  
}
```

Медиа запросы (@media) позволяют адаптировать страницу для различных типов устройств, таких как: принтеры, речевых браузеров, устройств Брайля, телевизоров и так далее.

В нашем случае используется медиа-функция max-width. Если ширина экрана меньше 650px, то некоторые CSS значения переопределяются:

- Убираются круглые края у шапки
- Элементы в main располагаются вертикально, из-за этого правая колонка смещается вниз
- Кнопки в навигационной панели и правой колонке располагаются вертикально

Структура сайта

```
└─ public
  └─ img
    └─ logo.svg
    └─ mail.svg
  └─ js
    └─ client.js
  └─ pages
    └─ journal
    └─ life
  └─ main
    └─ img
    └─ main.html
    └─ project
    └─ science
    └─ study
    └─ tech
    └─ style
  └─ favicon.ico
  └─ index.html
└─ package-lock.json
└─ server.js
```

В корневой директории содержится скрипт **server.js** – исходный код сервера на NodeJS.

В папке **public** содержатся файлы, отдаваемые клиенту.

Структура **public**:

- `img` – графика, используемая в шаблоне сайта.
- `js` – JavaScript сценарии.
- `style` – CSS стили, используемые в шаблоне
- `favicon.ico` – иконка, отображаемая на вкладке браузера
- `Index.html` – шаблон сайта

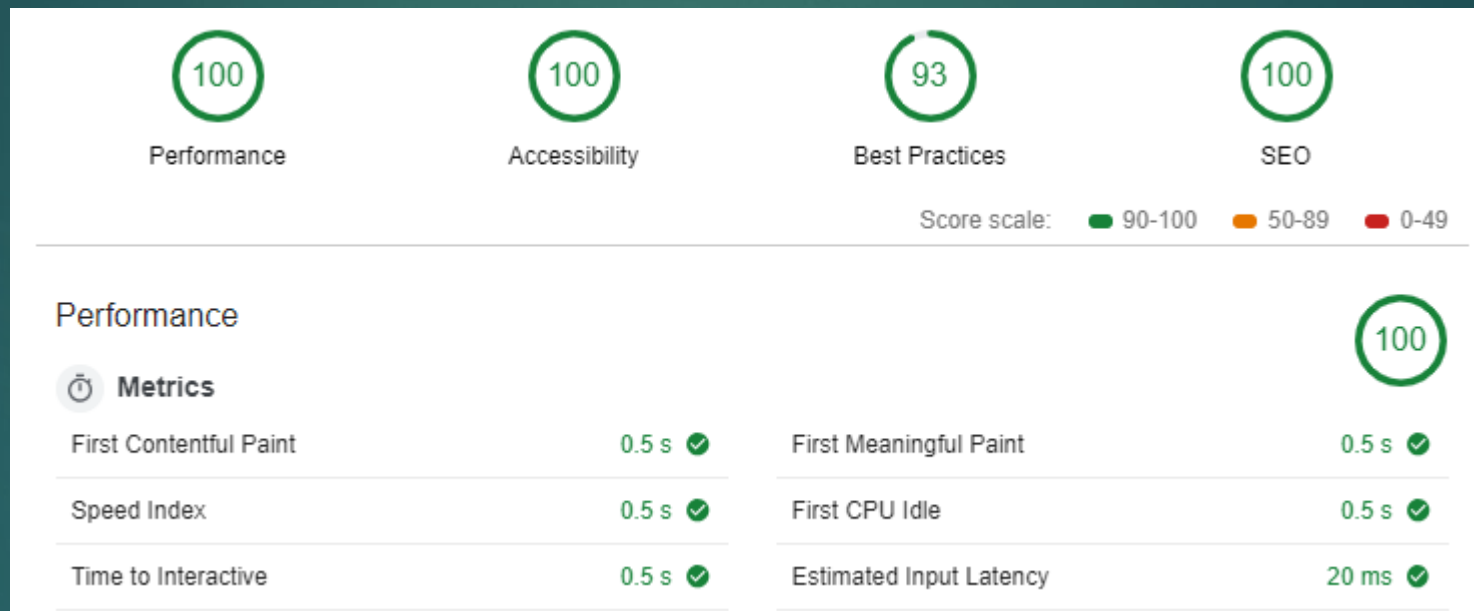
В **pages** хранятся подгружаемые страницы. Каждая папка содержит `html` файл представляющий вёрстку страницы и папку `img`, где хранятся изображения встраиваемые в страницу.

Использование REST API

В качестве сервера выступает NodeJS – программная платформа, позволяющая использовать язык JavaScript для разработки сервера. В качестве клиента разработан скрипт client.js.

1. При переходе пользователем по ссылкам в навигационной панели или сайдбаре скрипт client.js формирует POST запрос, с информацией о запрашиваемой странице.
2. Сервер обрабатывает запрос и отправляет html файл.
3. Клиент добавляет содержимое файла в тег article шаблона index.html.
4. Подгружаются изображения, ссылки на которые указаны в подгружаемом файле.

Соответствие современным стандартам



Lighthouse — проект с открытым кодом от Google, созданный для анализа производительности, дизайна и удобства использования сайтов. После аудита сервис формирует отчёт, в котором даёт рекомендации по улучшению сайта.

Произвести аудит можно в браузере Chrome, открыв панель разработчика (F12) на вкладке «Audits», выбрав интересующие метрики и нажав кнопку «Run Audits».

ИСПОЛЬЗОВАННЫЕ ИСТОЧНИКИ

- Чебатко М.И. “Варианты заданий и методические указания по выполнению лабораторных работ по дисциплине: «Мультимедийные системы»”. М. МАИ 2010
- <http://termins.evgmoskalenko.com/rus-r/rollover.html>
- <https://html5book.ru/css3-flexbox/>
- <http://htmlbook.ru/css/value/media>
- <https://ru.wikipedia.org/>