МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ  
Федеральное государственное бюджетное   
образовательное учреждение высшего образования   
ВЯТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Факультет автоматики и вычислительной техники  
Кафедра систем автоматизации управления

Методические указания   
по выполнению практического задания 2  
по дисциплине  
Web- технологии  
  
для студентов по направлениям подготовки   
09.03.02 Информационные системы и технологии

Киров 2023

# Цели и задачи

Цель работы**:** получение навыков компонентной разработки с использованием CSS библиотек.

# Порядок выполнения работы

## ***Изучить основные элементы и подходы к программированию в JavaScript***

При изучении языка JavaScript мы будем использовать интерактивный справочник <https://learn.javascript.ru/>, поэтому в дальнейшем будут лишь ссылки на его документацию для последовательного изучения материала.

Прочитайте введение в язык программирования <https://learn.javascript.ru/intro> и основания для его стандартизации <https://learn.javascript.ru/manuals-specifications>.

Существует 2 варианта работы с JavaScript - строгий, то есть без поддержки старых версий и совместимый <https://learn.javascript.ru/strict-mode>

Прочитайте об описании структуры кода <https://learn.javascript.ru/structure>, правилах наисания переменных <https://learn.javascript.ru/variables>, типах данных <https://learn.javascript.ru/types> и привдениях к ним <https://learn.javascript.ru/type-conversions>. Просмотрите базовые конструкции языка <https://learn.javascript.ru/operators> и операторы сравнения <https://learn.javascript.ru/comparison>, которые применяются во ветвлениях <https://learn.javascript.ru/ifelse>, <https://learn.javascript.ru/switch>   и циклах https://learn.javascript.ru/while-for .

Более сложный материал содержится про функции <https://learn.javascript.ru/function-basics>, так как появляются стрелочные функции <https://learn.javascript.ru/arrow-functions-basics> и обработке ошибок https://learn.javascript.ru/function-expressions .

В работе нам нужно провести аналогию с языком C# версии >8.

JavaScript - это ООЯП, поэтому нужно вспомнить что такое объекты <https://learn.javascript.ru/object> и как с ними работать <https://learn.javascript.ru/object-copy>, как к ним обращаться <https://learn.javascript.ru/object-methods> , как создавать <https://learn.javascript.ru/constructor-new> , как удалять https://learn.javascript.ru/garbage-collection , а также прочитать о типе данных Symbol <https://learn.javascript.ru/symbol>  и что такое опциональная цепочка <https://learn.javascript.ru/optional-chaining>

Изучите продвинутые методы работы JavaScript с функциями <https://learn.javascript.ru/advanced-functions>.

Этих знаний нам должно быть достаточно для понимания основ JavaScript.

## ***Изучить подходы к работе JavaScript в браузере***

Этих знаний нам должно быть достаточно для понимания основ JavaScript.

При работе с браузером мы должны прочитать что такое DOM-дерево <https://learn.javascript.ru/dom-nodes> и как работать с событиями <https://learn.javascript.ru/events> .

О разных типах событий браузера прочитать можно тут <https://learn.javascript.ru/event-details> , а если мы работаем с формами, то тут <https://learn.javascript.ru/forms-controls>

## ***Выполнить верстку макета по заданию преподавателя***

Изучение различных CSS атрибутов в современном мире JavaScript кажется не оптимальным занятием для разработчиков. Поэтому существуют подходы для систематизации и структуризации кода и разметки. Чтобы наглядно продемонстрировать данный подход Вам необходимо будет вычислить свой вариант по формуле: к каждой цифре своей зачетной книжки прибавить 6, и сложить полученный набор чисел между собой; затем разделить по модулю на 6, а к полученному результату прибавить 1.

Согласно Вашего варианта изучите макет который расположен по ссылке <https://drive.google.com/drive/folders/1FyPOUWChzrmuBRsgFXHbpFtW_0it3eb8?usp=sharing>. Возможно его нужно импортировать в графический редактор figma.

Необходимо реализовать данный макет с применением библиотеки tailwind <https://tailwindcss.com/>. Чтобы было удобнее установите помощник для вашего редактора кода <https://marketplace.visualstudio.com/items?itemName=bradlc.vscode-tailwindcss> .

Описание и мнение о библиотеке можно прочитать по ссылкам <https://maxsite.org/page/tailwind-css>  и <https://drbrain.ru/articles/tailwind-css/> .

Чтобы начать работу воспользуемся официальным руководством <https://tailwindcss.com/docs/installation#using-tailwind-via-cdn> и подключим библиотеку к шаблону как оно предлагает <https://tailwindcss.com/docs/installation#html-starter-template> .

Внимательно изучите принципы формирования названий css свойств в официальной документации <https://tailwindcss.com/docs/responsive-design>, стилизацию состояний  <https://tailwindcss.com/docs/hover-focus-and-other-states>, цаетовую палитру <https://tailwindcss.com/docs/customizing-colors>, отступы <https://tailwindcss.com/docs/customizing-spacing>.

Tailwind представляет собой удобный инструмент с огромным набором готовых шаблонов и механикой их оформления. Используйте набор иконок <https://github.com/aniftyco/awesome-tailwindcss>  если есть необходимость.

Для ускорения работы над версткой можно воспользоваться помощниками <https://tailblocks.cc/> и <https://tailwind.build/>, или же посмотреть свое вдохновение на сайте <https://tailwindcomponents.com/>

К полученной странице необходимо добавить на все кнопки интерактивное сообщение

Я это сделал, а @UserName тебе от меня совет: “ Поступай правильно — поступай в ВятГУ!”.

Вывод сообщения должен производится для вариантов 1 и 5 в виде promt, 2-3 в виде модального окна (https://sweetalert2.github.io/), а 4 и 6 - в форме всплывающего уведомления (<https://github.com/jaredreich/notie>). Ввод @UserName обязателен при первом выводе окна. Последующие выводы сообщений не должны требовать ввода @UserName.

По итоговому заданию необходимо написать пошаговую инструкцию как вы делали этот проект в формате отчета.

## ***Выполнить верстку макета по варианту***

Согласно варианта выполните верстку макета. Допускается использование CSS библиотеки tailwind css ()

Макет должен содержать следующие обработчики событий:

* перед отображением страницы на 543 миллисекунды должно отображаться сообщение по центру экрана “Загрузка страницы … “. Реализовать данную функцию можно с использованием компонента tailwind (<https://tailwindcomponents.com/component/full-page-overlay-loading-screen> или же так <https://codepen.io/damcclean/pen/GRoRRdR> )
* Все кнопки должны иметь модальные окна, стилизованные под общую тему оформления с сообщением “Макет был реализован {Ф.И.О. разработчика}. Вариант № {номер варианта}”.

Для работы с макетами используйте редактор figma.

Ссылка на макеты по вариантам представлена ниже:

<https://drive.google.com/drive/folders/1jx2fj_t3nT-VyoDtwYmb0j6gDR7ZtHeb?usp=sharing>.

Номер варианта соответствует номеру в названии макета

При проверке работы учитывается корректное отображение страницы в:

* браузер с полноэкранным разрешением 1920х1080;
* Apple iphone 12
* Google pixel 5
* Apple ipad mini