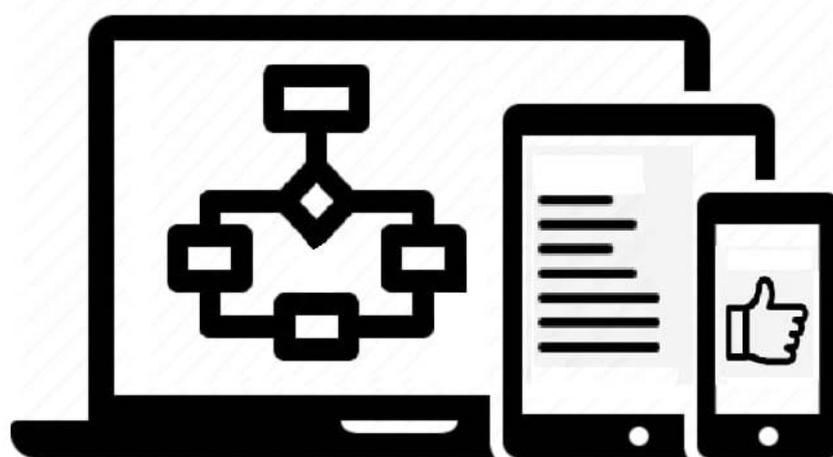


МЕЖГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«БЕЛОРУССКО-РОССИЙСКИЙ УНИВЕРСИТЕТ»

Кафедра «Программное обеспечение информационных технологий»

ОСНОВЫ WEB-ПРОГРАММИРОВАНИЯ

*Методические рекомендации к лабораторным работам
для студентов направления подготовки
01.03.04 «Прикладная математика»
очной формы обучения*



Могилев 2024

УДК 004
ББК 32.81
075

Рекомендовано к изданию
учебно-методическим отделом
Белорусско-Российского университета

Одобрено кафедрой «Программное обеспечение информационных технологий» «30» августа 2024 г., протокол № 1

Составитель канд. техн. наук Ю. В. Вайнилович

Рецензент канд. техн. наук С. К. Крутолевич

Методические рекомендации содержат требования к лабораторным работам по дисциплине «Основы web-программирования» для студентов направлений подготовки 01.03.04 «Прикладная математика» очной формы обучения.

Учебное издание

ОСНОВЫ WEB-ПРОГРАММИРОВАНИЯ

Ответственный за выпуск

В. В. Кутузов

Корректор

А. Т. Червинская

Компьютерная верстка

М. М. Дударева

Подписано в печать

. Формат 60×84/16. Бумага офсетная. Гарнитура Таймс.

Печать трафаретная. Усл. печ. л.

. Уч.-изд. л.

. Тираж 16 экз. Заказ №

Издатель и полиграфическое исполнение:

Межгосударственное образовательное учреждение высшего образования
«Белорусско-Российский университет».

Свидетельство о государственной регистрации издателя,
изготовителя, распространителя печатных изданий

№ 1/156 от 07.03.2019.

Пр-т Мира, 43, 212022, г. Могилев.

© Белорусско-Российский
университет, 2024

Содержание

Введение.....	4
1 Лабораторная работа № 1. Система контроля версий Git и веб-сервис GitHub	5
2 Лабораторная работа № 2. Фиксированная верстка готового дизайн-макета	6
3 Лабораторная работа № 3. Адаптивная верстка готового дизайн-макета. Технологии CSS Grid и Flexbox	8
4 Лабораторная работа № 4. Основы работы в Figma. Разработка дизайн-макета web-страницы.....	9
5 Лабораторная работа № 5. Адаптивная верстка дизайн-макета на основе CSS-фреймворка Bootstrap.....	9
6 Лабораторная работа № 6. Препроцессор SASS.....	9
7 Лабораторная работа № 7. Основы программирования на JavaScript..	10
8 Лабораторная работа № 8. Добавление интерактивности на web-страницу	11
9 Лабораторная работа № 9. Формы и модальные окна JavaScript.....	12
10 Лабораторная работа № 10. Асинхронный JavaScript.....	13
11 Лабораторная работа № 11. DOM Api	14
Список литературы	16
Приложение А. Варианты заданий для выполнения лабораторных работ ...	17

Введение

Цель учебной дисциплины состоит в формировании у студентов глубоких теоретических знаний и практических навыков в области веб-программирования, глубоком представлении об основных технологиях и инструментах, используемых при разработке веб-сайтов и приложений как на стороне клиента, так и на стороне сервера.

Методические рекомендации по дисциплине «Основы web-программирования» к лабораторным работам предназначены для оказания помощи студентам при выполнении лабораторных работ по данной дисциплине. Они содержат задания для самостоятельного выполнения, инструкции, пояснения и рекомендации, которые помогают студентам освоить конкретные технологии и принципы программирования, связанные с разработкой интернет-приложений.

По результатам выполнения каждой лабораторной работы студент предоставляет отчет, который содержит:

- титульный лист;
- цель работы;
- постановку задачи;
- описание результатов выполненной работы.

1 Лабораторная работа № 1. Система контроля версий Git и веб-сервис GitHub

Цель работы: освоить навыки работы с системой контроля версий Git.

Теоретический материал

Перед выполнением практического задания ознакомиться с [1].

Задание для самостоятельного выполнения.

Дана папка (скопировать у преподавателя), в которой хранятся файлы свеже созданного блога о диких животных. Файловая структура директории имеет вид:

```
wild_animals
├── index.html
├── pictures
│   ├── elephant.jpg
│   ├── giraffe.jpg
│   └── paw_print.jpg
```

В ходе задания будет необходимо создать на основе директории `wild_animals` репозиторий Git и внести в него некоторые изменения, после чего зафиксировать их средствами Git.

Ход выполнения задания.

1 Создайте репозиторий внутри папки `wild_animals`. Убедитесь, что внутри папки `wild_animals` появилась папка `.git`.

2 Настройте пользователя Git на уровне репозитория `wild_animals`:

- изучите содержимое файла конфигурации Git для текущего репозитория (`.git/config`);

- настройте имя и email пользователя для текущего репозитория;

- убедитесь, что файл `.git/config` изменился соответствующим образом;

3 Изучите содержимое папки `.git/objects`:

- убедитесь, что отсутствует файл индекса (`.git/index`);

- убедитесь, что папка с объектами Git пустая (`.git/objects`);

- убедитесь, что указатель HEAD указывает на ветку `main`;

4 Сделайте первый коммит:

- сделайте файлы папки `wild_animals` отслеживаемыми;

- обратите внимание на файл индекса (`.git/index`) и папку с объектами (`.git/objects`);

- сделайте коммит;

- найдите хэш коммита;

5 Сделайте второй коммит:

- исправьте опечатку в файле `index.html` (опечатка в слове `Elephant`);

- добавьте изменения в индекс;
- сделайте коммит;
- 6 Сделайте третий коммит:
 - добавьте в файл index.html секцию для еще одного животного (например, для кенгуру);
 - добавьте изменения в индекс;
 - сделайте коммит.

2 Лабораторная работа № 2. Фиксированная верстка готового дизайн-макета

Цель работы: освоить технологии HTML, CSS.

Теоретический материал

Изучить основы работы в Figma с использованием [1–4].

Ознакомиться со следующим материалом.

1 Основы CSS:

- <https://webref.ru/course/css-basics>.

2 Основы позиционирования элементов:

- <https://webref.ru/course/position>;
- <https://webref.ru/course/block-model>;
- <https://webref.ru/course/block-inline>.

3 CSS grid layout:

- основные понятия Grid Layout;
- полное руководство по CSS Grid;
- верстка на Grid в CSS;
- <https://css-tricks.com/snippets/css/complete-guide-grid>;
- <https://css-tricks.com/snippets/css/a-guide-to-flexbox/>.

4 Flexbox:

- <https://webref.ru/layout/flexbox-tutorial>;
- <https://habr.com/ru/post/467049/>;
- CSS: Flexbox;
- шпаргалка по Flexbox.

Задание для самостоятельного выполнения.

Разработать дизайн-макет браузерной версии лендинга. Сверстать спроектированный лендинг. Варианты тем лендингов выбираются из приложения А.

Требования к верстке

Верстка валидная. Для проверки валидности верстки используйте сервис <https://validator.w3.org/>. Валидной верстке соответствует надпись «Document checking completed. No errors or warnings to show.».

Верстка семантическая.

Запрещается использование CSS-фреймворков (bootstrap, foundation и т. д.). **Запрещается** использование JS-фреймворков (Angular, React, Vue и т. д.). **Запрещается** использование устаревших библиотек (jQuery и т. д.).

Для создания многоколоночных структур, или элементов, имеющих относительное горизонтальное расположение, должно быть использовано одно из свойств:

- display: flex;
- display: grid;
- display: inline-block.

Основные блоки должны быть точно расположены на заданной ширине экрана так, как в макете Figma.

Изображения, логотипы (если они есть) должны быть расположены в рамках логического контейнера с правильным подходом по центрированию и расположению. Допускается незначительное отклонение от макета в угоду сеточной или колоночной структуре.

Иконки, картинки должны сохранять идеальное расстояние до начала соответствующего им текста.

Иконки, картинки должны сохранять свои пропорции.

Если использован правильный шрифт, следует проверить высоту текста – он должен соответствовать исходнику. Ширина может варьироваться. Но общепринятой практикой является добавление свойства межбуквенного интервала (letter-spacing) тексту заголовков, девиза (motto) или цитат.

Если в строке несколько объектов визуальной одинаковой ширины, то ширина содержащих их блоков должна быть одинаковой. Разница размеров изображений не имеет значения, важно совпадение размеров блоков. Если в макете ширина блоков разная, то делать ее все равно нужно одинаковой.

Некоторые элементы должны быть интерактивными. «Интерактивный» означает, что у кнопки или элемента появляется визуальный эффект или анимация (на ваше усмотрение и исходя из макета: анимация курсора, изменение цвета заднего фона, затемнение, нижнее подчеркивание, изменение шрифта) при каких-либо действиях пользователя, например, при наведении курсора. Использовать JavaScript для обработки пользовательских событий в данном задании не обязательно. Обычно такой эффект реализуют при помощи псевдокласса: hover и следующих свойств:

- cursor: pointer;
- background;
- text-decoration: underline;
- color.

3 Лабораторная работа № 3. Адаптивная верстка готового дизайн-макета. Технологии CSS Grid и Flexbox

Цель работы: овладеть техниками и методологиями адаптивного дизайна, выработать умение применять их для создания гибких и отзывчивых веб-интерфейсов.

Теоретический материал

Ознакомиться со следующим материалом.

CSS3-медиазапросы (<https://html5book.ru/css3-mediazaprosy/>).

CSS3 Адаптивный веб дизайн. Медиазапросы (https://www.schoolsw3.com/css/css_rwd_mediaqueries.php).

Задание для самостоятельного выполнения.

Разработать дизайн-макеты для планшетной и мобильной версий лендинга. Сверстать спроектированные страницы.

Требования к верстке

Должны быть выполнены все требования лабораторной работы № 2 до порогового значения (меньше 320px). Это означает, что отступы, размеры блоков, и прочее, не должны уходить за правый край экрана и не должен появляться горизонтальный скролл.

Задание оценивается путем изменения размеров окна браузера Google Chrome или подключением эмуляции устройств через панель разработчика (DevTools -> Google Device Toolbar), выбрав значение ширины экрана.

При проверке должна отсутствовать вертикальная полоса прокрутки, т. к. она «съедает» часть пространства отзывчивой верстки своей шириной. Чтобы ее отключить, необходимо выбрать режим эмуляции Responsive, а также установить тип устройства Mobile (рисунок 1). Если тип устройства не отображается, в верхней панели device toolbar нажмите на три точки справа и выберите Add device type.



Рисунок 1 – Режим эмуляции

«Responsive» – это размеры, заданные в относительных величинах от ширины окна или родительского блока, которые плавно меняют свои значения при уменьшении или увеличении окна браузера. Главное, чтобы при наложении

картинки, например, в 768px на макет шириной 768px, размеры или отступы совпадали.

4 Лабораторная работа № 4. Основы работы в Figma. Разработка дизайн-макета web-страницы

Цель работы: разработать дизайн экранов, выбрать подходящие элементы управления; определить навигацию и создать прототипы интерфейса для браузерного приложения.

Теоретический материал

Перед выполнением практического задания ознакомиться с [1–6].

Задание для самостоятельного выполнения.

Разработать дизайн-макеты для 5–7 страниц десктопного, планшетного и мобильного вариантов веб-приложения.

5 Лабораторная работа № 5. Адаптивная верстка дизайн-макета на основе CSS-фреймворка Bootstrap

Цель работы: изучить и научиться применять компоненты и утилиты фреймворка Bootstrap для разработки веб-приложений.

Теоретический материал

Ознакомиться со следующим материалом.

Официальная документация по фреймворку Bootstrap (<https://getbootstrap.com/>).

Официальная документация по MaterialUI (<https://mui.com/material-ui/getting-started/>).

Задание для самостоятельного выполнения.

Для стилизации страниц веб-приложения использовать не менее 15 готовых компонентов, одну из библиотек компонентов на ваш выбор: Bootstrap или MaterialUI.

6 Лабораторная работа № 6. Препроцессор SASS

Цель работы: изучить и научиться применять препроцессор SASS для разработки веб-приложений.

Теоретический материал

Перед выполнением практического задания ознакомиться со следующим материалом.

Официальная документация (<https://sass-lang.com/>).

SASS: Основы SASS (<https://sass-scss.ru/guide/>).

Развернутое руководство по SASS/SCSS (<https://tproger.ru/translations/complete-sass-guide/>).

Задание для самостоятельного выполнения.

Переписать css-код лендинга, разработанного в лабораторной работе № 3, с использованием препроцессора SASS.

7 Лабораторная работа № 7. Основы программирования на JavaScript

Цель работы: разработать дизайн экранов, выбрать подходящие элементы управления, определить навигацию и создать прототипы интерфейса для мобильного приложения.

Теоретический материал

Ознакомиться со следующим материалом.

Учебник по JavaScript (<https://learn.javascript.ru/array-methods>).

15 методов работы с массивами в JavaScript, которые необходимо знать в 2020 году (<https://habr.com/ru/companies/plarium/articles/483958/>).

Документация по синтаксису JavaScript (<https://doka-guide.vercel.app/js/>).

Справочник по JavaScript. Встроенные объекты. Array (https://developer.mozilla.org/ru/docs/Web/JavaScript/Reference/Global_Objects/Array).

Задание для самостоятельного выполнения.

Задача 1. Реализовать функцию `check(str, bracketsConfig)`, которая для заданной последовательности скобок вернет `true`, если она правильная, и `false` – в противном случае.

Второй параметр `bracketsConfig` – массив пар открытых-закрытых скобок. Каждый подмассив включает в себя всего два элемента – открывающую и закрывающую скобки.

```
check('()', [['(', ')']]) // -> true
check('((( )))', [['(', ')']]) // -> true
check('()()', [['(', ')']]) // -> false
check('{{ }}', [['(', ')'], [['{', '}']]) // -> true
check('[ ]', [['(', ')'], [['[', ']']]) // -> false
check('[]()', [['(', ')'], [['[', ']']]) // -> true
check('[]()()', [['(', ')'], [['[', ']']]) // -> false
```

// особый случай: открывающая и закрывающая скобка могут быть одинаковыми

```
check('|', [['|', '|']) // -> true
check('|()', [['(', ')'], ['|', '|']) // -> true
check('|()', [['(', ')'], ['|', '|']) // -> false
check('|()|()', [['(', ')'], ['|', '|']) // -> true
```

Задача 2. Ваша задача – реализовать так называемый алгоритм «сортировки полотенца».

Функция TowelSort должна ожидать матрицу любой формы, например:

```
[
  [ 1, 2, 3 ],
  [ 4, 5, 6 ],
  [ 7, 8, 9 ],
]
```

Следующая матрица должна быть «отсортирована» по:

```
[1, 2, 3, 6, 5, 4, 7, 8, 9 ]
```

Задача 3. Ваша задача – написать функцию, которая декодирует азбуку Морзе и возвращает строку.

Длина строки ввода кратна 10.

Каждая буква алфавита закодирована точками (.) и тире (-). 10 означает точку (.), 11 означает тире (-).

Длина каждой закодированной буквы равна 10.

Если длина закодированной буквы меньше 10, она дополняется 0.

Пробел в строке обозначается «*****».

Вывод: строка (декодированная).

Пример: me -> m === -- === 0000001111, e === . === 0000000010 -> 00000011110000000010

8 Лабораторная работа № 8. Добавление интерактивности на web-страницу

Цель работы: добавить интерактивность ранее сверстанной веб-странице с использованием JavaScript для реализации указанного в задании функционала.

Теоретический материал

Ознакомиться со следующим материалом.

Учебник по JavaScript (<https://learn.javascript.ru/>).

Задание для самостоятельного выполнения.

Добавить для лендинга, сверстанного в лабораторной № 2, следующую интерактивность:

- перевод страницы на два языка;
- подсветка активной кнопки (например en/ru);

- возможность переключения «светлой» и «темной» тем;
- сохранить выбранные пользователем настройки в LocalStorage (язык отображения страницы и светлая или темная тема);
- смена изображения на странице;
- добавление звука;
- новые элементы должны быть стилизованы.

9 Лабораторная работа № 9. Формы и модальные окна JavaScript

Цель работы: научиться использовать формы и модальные окна для разработки веб-приложений.

Теоретический материал

Ознакомиться со следующим материалом.

Учебник по JavaScript (<https://learn.javascript.ru/array-methods>).

Document (<https://javascript.info/document>).

JavaScript DOM (<https://www.javascripttutorial.net/javascript-dom/>).

The HTML5 input types (https://developer.mozilla.org/en-US/docs/Learn/Forms/HTML5_input_types).

Руководство по HTML-формам. Валидация форм на стороне клиента (https://developer.mozilla.org/ru/docs/Learn/Forms/Form_validation).

Задание для самостоятельного выполнения.

Разработать страницу регистрации пользователя, включающую в себя:

- ввод номера телефона пользователя. Зарегистрироваться можно только на номер Беларуси;
- ввод email-пользователя;
- ввод даты рождения пользователя. Предусмотреть, что зарегистрироваться может пользователь, которому уже исполнилось 16 лет;
- возможность выбора способа задания пароля (самостоятельно или автоматически). Если пользователь вводит пароль самостоятельно, он его должен повторить (не используя вставку пароля в поле проверки). Пароль должен содержать минимум 8 символов, но не более 20; должен включать в себя хотя бы: одну заглавную букву, одну строчную букву, одну цифру и один специальный символ; не должен входить в TOP-100 самых распространенных паролей 2023 года;
- ввод ФИО пользователя. При этом отчество не является обязательным параметром;
- автогенерацию никнейма пользователя. Если пользователю автосгенерированный никнейм не нравится, предусмотреть возможность новой его генерации (по истечении пяти попыток необходимо предоставить возможность самостоятельного ввода никнейма);

- если никнейм пользователя уже существует, сообщить об этом;
- необходимость обязательного прочтения «Соглашения пользователя».

10 Лабораторная работа № 10. Асинхронный JavaScript

Цель работы: научиться получать данные от API и отображать их на веб-странице.

Теоретический материал

Ознакомиться со следующим материалом.

Using the Fetch API. ([https:// developer.mozilla.org/en-US/docs/ Web/API/ Fetch_API/Using_Fetch](https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API/Using_Fetch)).

Практическое ES6 руководство о том, как сделать HTTP запрос с помощью Fetch API. (<https://jem-space.ru/praktichieskoie-es6-rukovodstvo-o-tom-kak-sdielat-http-zapros-s-pomoshchiu-fetch-api/>).

Working with JSON. ([https:// developer.mozilla.org / en-US/docs / Learn / JavaScript/ Objects/JSON](https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Objects/JSON)).

Что такое JSON. (<https://habr.com/ru/post/554274/>).

Задание для самостоятельного выполнения.

Разработать приложение, отображающее полученные от API фото. Добавить приложению поиск. При вводе поискового запроса изменяются фото, которые отображаются в приложении.

Варианты API.

1 Unsplash API:

- сайт: <https://unsplash.com/developers>;
- документация: <https://unsplash.com/documentation>.

2 Flickr API:

- сайт: <https://www.flickr.com/services/>;
- документация: <https://www.flickr.com/services/api/flickr.photos.search.html>.

Требования к верстке

Верстка адаптивная. Приложение хорошо выглядит при ширине страницы от 1920px до 768px.

Интерактивность элементов, с которыми пользователи могут взаимодействовать, изменение внешнего вида самого элемента и состояния курсора при наведении, использование разных стилей для активного и неактивного состояния элемента, плавные анимации.

В футере приложения есть ссылка на GitHub автора приложения, год создания приложения.

Представленные варианты страниц не для копирования, а как пример того, что можно сделать.

Технические требования

Работа приложения проверяется в браузере Google Chrome последней версии.

Можно использовать bootstrap, material design, css-фреймворки, html и css препроцессоры.

Не разрешается использовать jQuery, другие js-библиотеки и фреймворки. JS-код приложения должен быть читаемым.

11 Лабораторная работа № 11. DOM Api

Цель работы: приобрести практические навыки навигации и управления DOM, понимание и применение CSS через JavaScript, а также обработки сложного поведения веб-документов.

Теоретический материал

Ознакомиться со следующим материалом.

Что такое Объектная Модель Документа (DOM)?

(https://developer.mozilla.org/ru/docs/Web/API/Document_Object_Model/Introduction).

Управление документами. (https://developer.mozilla.org/ru/docs/Learn/JavaScript/Client-side_web_APIs/Manipulating_documents).

Урок 7. JavaScript в веб-разработке (<https://smartiqa.ru/courses/web/lesson-7-js>).

Задание для самостоятельного выполнения.

Разработать виртуальную клавиатуру. Пример представлен на рисунке 2.

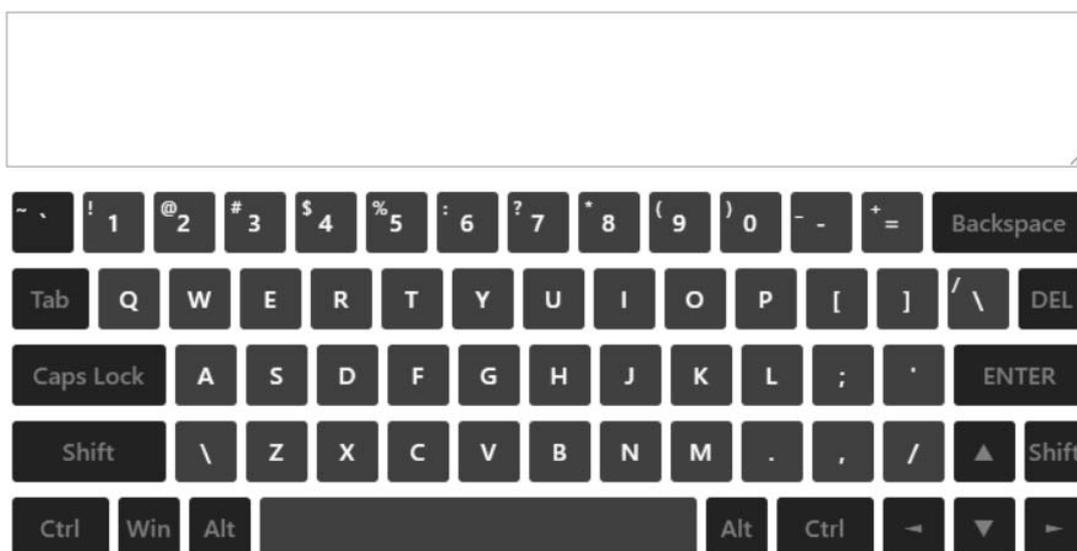


Рисунок 2 – Пример виртуальной клавиатуры

Требования к приложению

Дизайн на ваше усмотрение.

Index.html должен быть пустым. Все необходимые элементы генерируются с использованием JS.

Если нажато несколько кнопок, то на виртуальной клавиатуре подсвечиваются все нажатые кнопки (также нет исключений для Ctrl, Alt и Shift).

Виртуальная клавиатура способна переключаться между двумя языковыми раскладками (английский + любой другой язык).

Назначение сочетания клавиш для переключения раскладки клавиатуры зависит от вас.

Кнопки на виртуальной клавиатуре отображают символы выбранного языка.

Приложение сохраняет выбранный язык после перезагрузки страницы и отображает клавиатуру на этом языке.

На странице должна быть указана комбинация клавиш для смены языка, чтобы пользователю было понятно, как переключать раскладку клавиатуры.

Нажатия клавиш анимированы.

Щелчки мышью по кнопкам на виртуальной клавиатуре и нажатия клавиш на физической клавиатуре должны вводить символы в текстовую область, расположенную на странице над виртуальной клавиатурой.

Нажатие клавиши со стрелкой вверх, вниз, влево или вправо вводит символ стрелки в поле ввода или реализует навигацию по текстовой области.

Нажатие Enter должно переместить текстовый курсор на следующую строку.

Клавиша Tab создает горизонтальный отступ.

Нажатие остальных функциональных клавиш на клавиатуре не приводит к вводу символов.

Клавиша Backspace удаляет символ перед текстовым курсором.

Клавиша Del удаляет символ после текстового курсора.

Клавиши Shift, Alt, Ctrl, Caps Lock и Space должны работать как на настоящей клавиатуре.

Технические требования

Работа приложения проверяется в браузере Google Chrome последней версии.

Запрещается использовать Angular/React/Vue и другие фреймворки.

Не разрешается использовать jQuery, другие js-библиотеки.

JS-код приложения должен быть читаемым.

Список литературы

1 **Джейсон, Х.** Работа с Git. Полное руководство для начинающих и продвинутых пользователей / Х. Джейсон. – Москва: ДМК Пресс, 2018. – 544 с.

2 Figma – где скачать и как установить программу на компьютер? [Электронный ресурс]. – Режим доступа: <https://web-design.center/ustanovka-figma/>. Дата доступа: 20.09.2023.

3 **Окунев, А.** Руководство по Figma [Электронный ресурс] / А. Окунев. – Режим доступа: <https://medium.com/slashdesigner/figma-guide-5235b8a8ab4f>. – Дата доступа: 20.09.2023.

4 **Нагаева, И. А.** Основы web-дизайна. Методика проектирования: учебное пособие / И. А. Нагаева, А. Б. Фролов, И. А. Кузнецов. – Москва; Берлин: Директ-Медиа, 2021. – 236 с.

5 Figma с нуля – основы работы с Фигмой для веб-разработчика, верстальщика и дизайнера. Полный обзор [Электронный ресурс]. – Режим доступа: <https://www.youtube.com/watch?v=z6mlqOGmjQQ>. – Дата доступа: 20.09.2023.

6 Лекция 3: Основы разработки интерфейсов мобильных приложений интерфейсов [Электронный ресурс]. – Режим доступа: <https://intuit.ru/studies/courses/12643/1191/lecture/21986?page=1>. – Дата доступа: 20.09.2023.

7 Типы адаптивных макетов [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/post/158703/>. – Дата доступа: 20.09.2023.

Приложение А (справочное)

Варианты заданий для выполнения лабораторных работ

- 1 Автострахование.
- 2 Агентство по сдаче автомобилей в аренду.
- 3 Аренда коньков, роликов, велосипедов, лыж.
- 4 Аэропорт: пассажирское расписание и перевозки.
- 5 Банковская система вкладов (физических и юридических лиц).
- 6 Банковская система кредитования (физических и юридических лиц).
- 7 Биллинг сотовой компании.
- 8 Ветеринарная лечебница.
- 9 Клуб обучения танцам.
- 10 Магазин косметики.
- 11 Машиностроительное предприятие: система по разработке и модификации изделий (ведение архива, стандартов и пр.).
- 12 Нефтеперерабатывающая компания.
- 13 Парикмахерская.
- 14 Поставка вин.
- 15 Приемная комиссия вуза.
- 16 Производство мебели (прием индивидуальных и типовых заказов и изготовление).
- 17 Рекламное агентство.
- 18 Риелторская компания: аренда, продажа первичного и вторичного жилья.
- 19 Санаторий.
- 20 Система управления проектом для IT-компании.
- 21 Складская логистика.
- 22 Спа-салон (услуги, обслуживающий персонал и пр.).
- 23 Страховая компания.
- 24 Такси.
- 25 Транспортная логистика.
- 26 Туристическое агентство (путешествия за рубеж).
- 27 Туристическое агентство (путешествия по Беларуси).
- 28 Учет оборудования на крупном промышленном предприятии.
- 29 Филармония.
- 30 Электронный проездной.