**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**Учреждение образования**

**«Гомельский Государственный**

**Технический Университет**

**имени П.О. Сухого»**

**Кафедра «Информационные Технологии»**

**Лабораторный практикум**

**по дисциплине**

«СОВРЕМЕННЫЕ ТЕХНОЛОГИИ FRONT-END РАЗРАБОТКИ»

для специальности

"информационные системы и технологии"

Гомель 2021

**Лабораторная работа №1:**

**«Front-End фреймворки»**

Разработать интерфейс веб-приложения с использованием *frontend*-фреймворков *React*, *Angular* или *Vue* (на выбор).

Темы веб-приложений использовать из таблицы согласно варианту:

|  |  |
| --- | --- |
| Вариант | Описание приложения |
| 1 | Курсы валют. Отображать отношение BYN к USD, RUB, EUR. Иметь форму для расчёта конвертации валют (перевод валют по курсу). |
| 2 | Время. Отображать на странице время в определённой стране мира по указанному пользователем названию страны. Время должно динамически меняться в режиме реального времени. |
| 3 | Перевод единиц длин. Предоставлять пользователю указать две величины длинны: исходную и ту, в которую надо конвертировать (например, километры в метры, футы в дюймы, мили в сантиметры). |
| 4 | Конвертация HEX-кода [пример: “#RRGGBB”] цвета в его RGB-эквивалент [пример: “rgb(RRR,GGG,BBB)”]. |
| 5 | Перевод единиц масс. Предоставлять пользователю указать две величины массы: исходную и ту, в которую надо конвертировать (например, килограммы в граммы, фунты в унции). |
| 6 | Переводчик. Перевод текста с одного языка на другой. Иметь возможность перевода на/с не менее пяти языков. |
| 7 | Перевод объёма информации. Предоставлять пользователю указать две величины объёма информации: исходную и ту, в которую надо конвертировать (например, байты в килобайты, гигабайты в мегабайты, терабайты в гигабайты). |
| 8 | Местоположение по IP. Определить город по заданному IP-адресу. |
| 9 | Перевод единиц температур. Предоставлять пользователю указать две величины температуры: исходную и ту, в которую надо конвертировать (например, градусы Цельсия в Фаренгейта, Кельвины в градусы Цельсия) |
| 10 | Чат-бот типа «вопрос/ответ». В интерфейсе должно иметься текстовое поле для ввода строгих команд (без вариаций формулировки и проверки правописания, простая строгая команда), на которые заранее заготовлены определённые ответы, соответствующие этим командам. |
| 11 | Словарь стран и столиц. Определить страну по столице и наоборот. В интерфейсе иметь только одно текстовое поле для ввода и названия страны, и его же использовать для ввода названия столицы. Не использовать переключатели. Если в текстовом поле указано название страны – отобразить название её столицы. Если в этом текстовом поле указано название столицы – отобразить название страны, страны, столицей которой она является. |
| 12 | Перевод единиц времени. Предоставлять пользователю указать две величины времени: исходную и ту, в которую надо конвертировать (например, часы в секунды, года в недели, месяца в минуты и т.д.). |

В зависимости от задачи, можно как создавать собственные словари / шаблоны, так и пользоваться общедоступными бесплатными API / json различных сервисов.

Разработанное приложение должно иметь:

• Шапку с логотипом (произвольным, можно текстовым) и темой работы;

• Футер (с подписью студента, который разрабатывал веб-приложение);

• Основная область веб-приложения.

**Лабораторная работа №2:**

**«Адаптивная вёрстка»**

2.1 При необходимости, переписать стили с использованием современных единиц измерения для отзывчивой вёрстки, используя макеты *flex* и/или *grid*, которые очень упрощают создание адаптивных макетов. При необходимости, допускается использование *Bootstrap* (не ниже 4-ой версии).

2.2 Адаптировать разработанное в прошлой лабораторной работе веб-приложение под мобильные устройства, используя медиа-запросы или подключение стилей с использованием медиа-условий.

2.3 Разработать «ночную тему» оформления, основываясь на установленной в системе мобильного устройства темы оформления или настроек браузера.

2.4 При повороте всего интерфейса мобильного приложения в ландшафтный режим отобразить «шторку» с уведомлением о том, что веб-приложение может отображаться только в альбомном режиме. «Шторка» должна скрывать под собой весь основной контент до тех пор, пока устройство не повернут в альбомный режим.

**Лабораторная работа №3:**

**«Взаимодействие с API браузера»**

* 1. В приложении из ЛР№1 добавить меню для отображения основного контента из работы №3.
  2. Реализовать в уже имеющемся приложении функционал согласно варианту из приведённой ниже таблицы:

|  |  |
| --- | --- |
| Вариант | Описание функционала |
| 1 | Используя HTMLMediaElement, Web Audio API и, при необходимости, WebRTC, разработать музыкальный audio- или video-плеер кастомной панелью управления: то есть стандартный «барузерский» контроллер управления аудио-файлом нужно убрать и сделать неактивным, а для управления аудио/видео-файлом (переключать предыдущий/следующий треки, перематывать вперед/назад, делать громче/тише, ставить на паузу / воспроизводить) необходимо разработать собственный интерфейс. Также реализовать возможность с помощью клавиш управлять регулировкой уровня звука (стрелки «вверх» и «вниз») и перемоткой на N-секунд (стрелки «влево» и «вправо»).  По завершению воспроизведения одного аудио/видео файла, автоматически воспроизводить следующий.  По нажатию на любой элемент управления плеером (кнопки перемотки, регулировкой звука, переключения трека/видео), вызывать один вибро-сигнал (вибрацией устройства можно управлять с помощью *Vibration API*). |
| 2 | На основе Geolocation API мобильного девайса, определить его местоположение и отобразить метку его точки местонахождения на карте (допускается использовать любой сервис, вроде Google.Maps, Yandex.Maps, OpenStreetMap, 2Gis и т.п.). Если на устройстве отключена функция определения точного местоположения, необходимо отобразить соответствующее уведомление на экране и возобновить работу после его включения.  Реализовать возможность вручную добавить маркер второй точки и вычислить между ними (вручную добавленной автоматически определённой) расстояния (в метрах).  При определении местоположения устройства совершать три вибро-сигнала (вибрацией устройства можно управлять с помощью *Vibration API*). |
| 3 | Используя браузерное хранилище устройства (например, *Locale Storage*), разработать TODO-лист для различных назначений (например, для организации списка задач), реализовав интерфейс для CRUD-операций (создание новой записи, отображения записи/записей, его/их редактирования и удаления). Список TODO-листа должен быть актуальный даже после перезагрузки страницы и закрытия браузера с веб-приложением (то есть его состояние и содержимое при следующем открытии должно быть такое, каким было при закрытии страницы).  При успешном выполнении операции добавления новой записи, сохранения записи после редактирования или удаления записи, заставлять мобильное устройство совершить отклик в виде двух краткосрочных выбиро-сигналов (вибрацией устройства можно управлять с помощью *Vibration API*). |
| 4 | Используя браузерное API для рисования *Canvas API*, реализовать интерфейс для передвижения условной фигуры (любой кроме квадрата!) по полю. Движение фигуры должно осуществляться автоматически, а её направление должно меняться по нажатию на стрелки клавиатуры на ПК и с помощью «свайпа» (движения пальцем по экрану) на мобильных устройствах.  На выбор реализовать один из дополнительных возможностей:  - увеличение фигуры от «съедания» случайно появившихся объектов на полотне (при соприкосновении объекта фигуры с объектом для увеличения);  - окончания игрового процесса (и запуск с начала) при столкновении с фигуры с границами поля (при этом скорость движения фигуры должна постепенно расти);  - окончание игрового процесса при столкновении с движущимся по полю объектом / объектами.  При выполнении выбранного условия, заставлять мобильное устройство совершить два краткосрочных вибро-сигнала (вибрацией устройства можно управлять с помощью *Vibration API*). |