1. Основы HTML. Работа с ссылками. Работа с изображениями (7 баллов)
2. Основы CSS. Блоки. Спрайты (7 баллов)
3. Адаптивная верстка (@media) (8 баллов)
4. JS. Основы. Слайдер (8 баллов)
5. Асинхронность. Работа с https://jsonplaceholder.typicode.com/ (10 баллов)
6. Шаблонизаторы. Handlebar (или что-то подобное) (10 баллов)
7. Основы React. Перенос страничек на React. (10 баллов)

ДЗ -

С использованием React разработать приложение, используя открытое API pokeapi.co.

Требования к приложению:
1. Приложение должно содержать карточки с информацией о покемонах.
2. Приложение должно иметь несколько страниц (более 3-х).
3. Приложение должно содержать список избранных покемонов, который должен сохраняться при перезагрузке страницы.
4. Приложение должно содержать функцию поиска покемонов (если API не позволяет, можно не реализовывать).
5. Запросы к API должны выполняться при помощи библиотеки Axios.
6. Приложение должно быть разбито на компоненты.
7. Использование модулей обязательно.

При разработке приложения можно выбрать любое API и тему. По согласованию с преподавателем и с сохранением функционала описанного выше.

1. Основы HTML. Работа с ссылками. Работа с изображениями

Разработать страницу содержащую в себе

1. Общий заголовок
2. Список якорных ссылок на разделы (Реализовать в виде списка ul)
3. Раздел о себе
4. Раздел навыков
5. Раздел любимых сайтов

Каждый раздел должен содержать

1. Подзаголовок
2. Картинку
3. Текст с форматированием по желанию

Текст должен быть стилизован. Для этого можно использовать как специальные HTML теги так и inline стили пример.

Разделы должны быть разделены линией.

2. Основы CSS. Блоки. Спрайты.

Разработать страницу, содержащую карточки с информацию о ваших одногруппниках (от 4-х до 8). Каждая карточка должна содержать Имя, Фамилию, Фото и поля по вашему усмотрению. Карточки должны располагаться в нескольких столбиках. Для стилизации использовать CSS, стилевые правила должны быть описаны в отдельном файле, использовать классы. Также поменять inline стили на первой странице на CSS правила описанные в отдельном файле. Обязательно должны быть изменены шрифт и размер шрифта, а также стили карточек.

Добавить шапку на страницах, с ссылками на 1-ю и 2-е лабораторные работы. Также добавить ссылку на любую соцсеть. Ссылку сделать в виде картинки, при наведении на которую будут

меняться состояния. Смену состояний сделать в виде спрайта.

3. Адаптивная верстка

Сделать так, чтобы текст был читабельным на страницах разработанных в ходе первой и второй лабораторных работ на различных устройствах (мобильные устройства и планшеты). Размер шрифтов вынести в переменные.

4. JS. Основы. Слайдер

Создать отдельную страницу содержащую:

1. Шапку
2. Слайдер (Разработанный при помощи CSS, JS, HTML. Стили и скрипт подключить из отдельного файла. Картинки выводить динамически (не должны быть описаны в HTML файле))

5. Асинхронность.

При помощи fetch запроса сделать запрос к API - https://jsonplaceholder.typicode.com/ . Вывести полученный результат на страницу. Во время загрузки данных вывести надпись Loading… в случае ошибки вывести Network error, try again later.

Добавить стили на полученную страницу.

6. Шаблонизаторы

Выбрать любой сборщик проектов и вынести шапку и подвал в отдельные компоненты которые в дальнейшем использовать на всех страницах.

7. Основы React

Разработайте простое веб-приложение с использованием React, в котором будет реализована функциональность отображения списка задач.

Требования к приложению:
1. Создайте компонент TaskList, который будет отображать список задач. Каждая задача должна быть представлена отдельным компонентом Task.
2. Реализуйте возможность добавления новых задач в список. Для этого создайте компонент AddTaskForm, который будет содержать поле ввода для названия задачи и кнопку "Добавить".
3. Реализуйте возможность удаления задачи из списка. Для этого добавьте кнопку "Удалить" к каждой задаче в компоненте Task.
4. Добавьте возможность отметить задачу выполненной. Для этого добавьте чекбокс к каждой задаче в компоненте Task.
5. При отметке задачи выполненной, измените ее стиль или цвет, чтобы пользователь мог легко видеть выполненные задачи.

Дополнительные задания:
1. Добавьте возможность редактирования названия задачи. Для этого добавьте поле ввода и кнопку "Сохранить" к каждой задаче в компоненте Task.
2. Реализуйте фильтрацию списка задач по выполненности. Добавьте компонент Filter, который будет содержать чекбокс "Показать выполненные задачи" и "Показать невыполненные задачи". При выборе одного из вариантов, должны отображаться только соответствующие задачи.
3. Добавьте возможность сохранения списка задач в локальное хранилище браузера, чтобы при обновлении страницы задачи не пропадали.