



Марина Слукина

26 лет

frontend-разработчик

Контакты

Телефон: +79052583993

Почта: slukinamarina@gmail.com

Портфолио:

<https://github.com/Marinaslukina/portfolio>

Высшее образование

2020 – Университет ИТМО

Факультет систем управления и робототехники

Направление подготовки: Системы управления движением и навигация

- программа для оценки температурной погрешности гироскопа на C++
- программа для прогнозирования качки корабля в Matlab

Дополнительное образование

2020 – н. в. Skillbox

Frontend-разработчик

- пройден курс Javascript «Базовый уровень»
 - 1 Разработка простой игры в «пары». Задача: игрок видит квадратное поле из карточек, расположенных «рубашкой» вверх, и находит пары одинаковых карточек, открывая их в произвольном порядке. Перед игрой можно настроить количество карточек на поле с помощью формы. Игра начинается нажатием на кнопку. Также есть таймер в 1 минуту, по истечению которого игра сразу завершается, даже если еще не открыты все карточки.
 - 2 Создание панели управления студентами. Задача: создать панель, на которой будут располагаться таблица со студентами с фильтрами и сортировкой, форма

добавления нового студента.

Студенты хранятся в массиве. Каждый студент – объект. Для добавления студентов на странице есть форма с полями фио, дата рождения, год начала обучения, факультет.

Форма проходит валидацию по следующим правилам:

- все поля обязательны для заполнения после применения к значению метода trim();
- дата рождения находится в диапазоне от 01.01.1900 до текущей даты;
- год начала обучения находится в диапазоне от 2000 до текущей даты.

Валидация происходит после нажатия на кнопку «добавить студента». Если валидация прошла успешно, новый студент добавляется в таблицу, все поля очищаются. В противном случае выводятся сообщения об ошибках.

Данные из массива выводятся в табличном виде. Каждая строка таблицы содержит информацию об студенте:

- фио;
- факультет;
- дата рождения и возраст в формате «31.12.2000 (22 года)»;
- годы обучения и номер курса в формате «2019-2023 (4 курс)». Считается, что все студенты учатся 4 года. Если сентябрь года окончания уже прошел, то выводится «закончил».

Первая строка таблицы заголовочная. При нажатии на ячейку заголовочной строки происходит сортировка по соответствующим полям:

- фио сортирует по соединенной строке из фамилии, имени и отчеству по алфавиту по возрастанию;
- факультет – по факультету по алфавиту по возрастанию;
- ДР и возраст – по дате рождения по возрастанию;
- годы обучения – по году начала обучения.

Перед таблицей есть фильтр из полей:

- фио для поиска подстроки в фамилии, имени или отчестве;
- факультета для поиска подстроки в названии факультета;
- год начала обучения (точное совпадение);
- год окончания обучения (точное совпадение).

Использовала: bootstrap

- 3 Список статей блога. Задача: создание блога из двух страниц (двух html файлов) на основании <https://gorest.co.in/>.

Первая страница: это список статей блога, который можно получить запросом GET. Номер страницы берется из URL страницы из параметра page. На странице выводится:

- список статей, где каждый элемент – ссылка на статью;
- постраничная навигация. Кнопки «<<», «<», «>», «>>».

Детальная страница статьи получается запросом <https://gorest.co.in/public-api/posts/{id статьи}>. Id статьи берется из URL страницы из параметра id. На странице выводится:

- заголовок из ответа на запрос;
- содержимое из ответа на запрос;
- комментарии к статье в виде списка с именем автора и содержимым к каждому комментарию. Список комментариев к конкретной статье можно получить запросом GET https://gorest.co.in/public-api/comments?post_id=4.

Использовала: bootstrap

- пройдено 30% курса Javascript «Продвинутый уровень»

- 4 Блок, выпадающий вниз при нажатии на кнопку. Задача: блок должен закрываться при нажатии в любом месте за его пределами. Для реализации

нельзя было использовать `stopPropagation`.

Содержимое выпадающего блока не имеет значение, но любое нажатие мыши внутри этого блока не должно приводить к его закрытию.

Использовала: `bootstrap`

- 5 Обработка событий с клавиатуры и валидации полей для ввода. Задача: создать форму с полями для ввода ФИО. В каждом поле допускается только ввод кириллицы в любом регистре, дефиса и пробела. При вводе с клавиатуры любая кнопка, не соответствующая допустимому символу, не приводит к вводу символа в поле.

При потере фокуса происходит проверка на корректность введенного значения и замена его на корректное значение при необходимости по следующим правилам:

- из строки значения удаляются все символы, кроме допустимых;
- пробелы и дефисы в начале и конце значения удаляются;
- несколько идущих подряд пробелов или дефисов заменяются на один;
- первая буква приводится к верхнему регистру, а все остальные — к нижнему.

Отправка формы добавляет абзац с введенными ФИО под форму без перезагрузки страницы. Каждая отправка формы добавляет новый абзац на страницу. После отправки поля на форме очищаются.

Использовала `bootstrap`

- 6 Прокрутка. Задача: создание кнопки «наверх», которая появляется только при прокрутке >100 пикселей. Нажатие на кнопку плавно прокручивает страницу вверх. Содержимое страницы не имеет значение.
- 7 Переключение хранилища в приложении TODO. Задача: в ранее созданном приложении TODO добавить кнопку переключения хранилища. Если использовалось локальное хранилище браузера, то

происходит переключение на сервер с API, и наоборот. Разделение приложения на модули. Использовала bootstrap

- 8 Работа с асинхронными вызовами. Задача: создать каталог эпизодов Star Wars с выводом детальной информации о них с помощью <https://swapi.dev/>. Подробности одного эпизода можно получить запросом GET <https://swapi.dev/api/films/{номер эпизода}>. Приложение имеет 2 страницы:

- главная страница. На ней загружается и показывается список эпизодов с указанием номера и названия в каждом элементе. Элементы должны быть ссылками на детальную страницу эпизода;
- детальная страница эпизода должна загружать и показывать информацию об эпизоде, номер которого должен быть передан в URL страницы. На детальной странице должны быть следующие элементы: название эпизода и его id; кнопка «Back to episodes» со ссылкой на список эпизодов (главную); описание эпизода; список планет (можно из API с использованием URL из массива planets в объекте эпизода); список рас (можно из API с использованием URL из массива species в объекте эпизода).

- более 50% курса Веб-верстка «Базовый уровень»
- пройден курс система контроля версий Git

2020 - Учебный центр **Netcracker Technology**, стажировка

Quality Assurance

- ручное тестирование сайта РЖД с последующим заведением тест кейсов, тест ранов и багов в JIRA
- составление тест стратегии, тест плана, тест кейсов для тестирования лифта
- тестирование дизайн спецификации и функциональных требований для приложения Notebook с последующим

заведением тест кейсов

- работа с БД при помощи основных команд в SQL
- запуск и отладка Selenium ide Chrome тестов

Опыт работы

2018 – н. в. АО «Концерн «ЦНИИ «Электроприбор»,

Санкт-Петербург

Инженер-разработчик

- разработка, корректировка, согласование и выпуск эксплуатационной документации для инерциальных систем навигации и стабилизации надводных кораблей
- ведение деловой переписки внутри предприятия и с контрагентами
- участие в испытаниях

Технические навыки

HTML, CSS, Bootstrap, JavaScript, Adobe Photoshop, Figma, JIRA, GIT

О себе

- выполняю задачи в установленные сроки
- готова учиться и развиваться
- английский язык чтение и письмо
- немецкий язык чтение и письмо