



Алымбаева Асель Кадырбековна

Front-end developer

место положение: **Москва**

локация: **Да**

Hard skills

- JavaScript
- TypeScript
- React
- HTML
- CSS
- Webpack
- SCSS, SAAS
- Git
- Eslint
- Perfect Pixel
- Figma
- NPM, YARN

Soft skills

- Trello
- Miro
- KanBan
- Canva

Language

- English
Pre-Intermediate

Contacts

- tel: 8 925 533 32 50
- mail:
asel.alimbaeva.kk@gmail.com
- discord:
asel.alimbaeva.kk(alimbaeva)(kg)
- Telegram
@aselAI86
- GitHub
@alimbaeva

Образование

КГУ им. И. Арабаева - сурдопедагог со специальной психологией

Курсы

● Jyldyz Academi

Front-end developer - 05.05.21 - 30.12.21

● RSSchool

Front-end developer - 05.12.21 - 13.09.22

● RSSchool

React - 21.09.22

Опыт работы

Учебные командные проекты:

RSLang: TS

ссылка: <https://github.com/rrroeva/rslang>

ссылка на сайт: <https://rolling-scopes-school.github.io/rrroeva-JSFE2022Q1/rslang/>

ссылка на макет: <https://www.figma.com/file/sqgbTWnOaWy7d3O6yK9aFN/RS-LANG?node-id=0%3A1>

Моя задача:

- вёрстка, дизайн, UI страниц электронного учебника
- электронный учебник генерируется на основе коллекции исходных данных и состоит из шести разделов, в каждом разделе 30 страниц, на каждой странице 20 слов для изучения
- седьмой раздел учебника - "Сложные слова" изначально пустой. Этот раздел состоит из слов, которые пользователь отметил как сложные. Все слова в этом разделе размещаются на одной странице. На усмотрение разработчиков, слова в данный раздел добавляются либо в том порядке, в котором пользователь отмечал их как сложные, либо в порядке, в котором они находились в учебнике.
- на каждой странице учебника отображается:
 - меню или иконка меню
 - список из 20 слов (в разделе "Сложные слова" слов может быть больше)
 - ссылки на мини-игры "Аудиовызов" и "Спринт"
 - навигация по страницам учебника
 - также необходимо реализовать навигацию по семи разделам учебника и предусмотреть небольшие различия в оформлении каждого раздела
 - при перезагрузке страницы открывается последняя открытая страница приложения

Список слов +80

- вёрстка, дизайн, UI списка слов
- для каждого слова отображается:
 - само слово, транскрипция, перевод
 - предложение с объяснением значения слова, перевод предложения
 - предложение с примером использования изучаемого слова, перевод предложения
 - картинка-ассоциация к изучаемому слову
 - иконка аудио, при клике по которой последовательно звучит произношение изучаемого слова, произношение предложения с объяснением его значения, произношение предложения с примером его использования
- только у авторизованных пользователей отображаются:
 - кнопка, при клике по которой слово можно отметить как сложное (в разделе "Сложные слова" на её месте отображается кнопка, снимающая отметку что это сложное слово и удаляющая его из данного раздела)
 - кнопка, при клике по которой слово можно отметить как изученное
 - если слово отмечено как сложное, оно остаётся на странице учебника и выделяется стилем, указывающим, что данное слово относится к сложным словам. Также данное слово добавляется в раздел "Сложные слова"
 - если слово отмечено как изученное, оно остаётся на странице учебника и выделяется стилем, указывающим, что данное слово относится к изученным словам
 - если все слова на странице относятся к изученным словам или к изученным и сложным словам, такая страница считается полностью изученной и выделяется стилем. Также стилем выделяется соответствующая ей кнопка навигации по страницам учебника (достаточно выделить номер открытой в данный момент страницы, если она полностью изучена). Ссылки на мини-игры на этой странице становятся неактивными.

Project Management System: React TS MUI

ссылка на PR: <https://github.com/Darina992/project-management-app/pull/26>

deploy: <https://darina992-project-managment-app.netlify.app/>

Моя задача:

- Создать новую доску - открывает модальное окно с формой для создания доски. Требования к форме такие же, как и ко всем формам в заявке.
- Отображает все созданные доски в виде списка.
- Доски отображаются с небольшим предварительным просмотром доступной информации (название, описание и т.д.). Нажав на элемент, пользователь перенаправляется на элемент доски (маршрут доски). Должна быть кнопка для удаления доски.
- Когда пользователь пытается удалить доску, он / она должен получить модальное подтверждение, чтобы проверить, хочет ли пользователь удалить доску (чтобы избежать удаления доски по ошибке). Способ подтверждения должен быть общим компонентом (один для приложения).
- Пример дополнительной функциональности (необязательно): глобальный поиск. Поиск задачи по номеру задачи, названию, назначенным ей пользователям и по тексту описания задачи.
- Реализована функциональность просмотра и редактирования задачи.
- Задача должна иметь кнопку удаления задачи. По щелчку: режим подтверждения
- Задача имеет информацию о создателе и данные доски и колонки

redux для создания столбца

ООО СТК - бригадир цеха

Моя задача:

- Управление бригадой, направление и разделение работы и обязанности между работниками
- Прием и заказ продукции и проверка качества товаров.