**Требования к проекту**

**1. Введение**

Цель разработки – создать одностраничное веб-приложение TodoApp, которое позволит пользователям сохранять свои нужные дела, а также отмечать выполненные и следить за прогрессом

**2. Требования пользователя**

2.1 Программные интерфейсы

Разрабатываемый проект будет сделан в виде веб-страницы в браузере при помощи библиотеки React JS + NodeJS в качестве бэкенда.

2.2 Интерфейс пользователя

При запуске приложения пользователь увидит веб-страницу с полем для ввода, после ввода в которое и нажатия на клавишу в список занесется задача. Задачу можно будет убрать с помощью галочки, а также вывести список всех задач пользователя.

2.3 Характеристики пользователей

Так как никто не застрахован от забывчивости, приложение пригодится всем возрастным группам, особенно тем, у кого на работе есть персональный компьютер, в который по мере необходимости можно вносить свои дела, чтоб не забыть. Также понятный интерфейс поможет людям, кто нечасто общается с компьютером.

Целевая аудитория – люди от 15 до 75, т.е. все, у кого есть ПК.

2.4 Предположения и зависимости

Возможна некоторая смена отображаемости в разных браузерах, плюс старые браузеры не будут поддерживать современную библиотеку.

**3. Системные требования**

Для запуска приложения необходим персональный компьютер с установленным одним из любых браузеров на выбор пользователя.

3.1 Функциональные требования

1) При нажатие на поле для ввода дать возможность пользователю ввести с клавиатуры текст для новой задачи.

2) При нажатии пользователем на кнопку “Add todo” занести новую задачу в список всех задач и отобразить на экране без перезагрузки страницы.

3) При нажатии пользователем «галочки» возле любой задачи программа должна скрывать задачу из списка как выполненную.

4) При вводе пользователем текста в поле “Search” программа должна искать те задачи, в названии которых есть данные символы, а затем отобразить только то, что совпадает.

5) При нажатии на checkbox с именем “Show completed todos” программа отображает те задачи, которые пользователь отметил как выполненные. Выполненные задачи необходимо стилизовать по другому(текст у них другого цвета, и сами они перечеркнуты).

6) Предусмотреть сохранение пользовательских данных, т.е. когда

пользователь перезапустит приложение, он увидит свой список дел.

7) Задача содержит в себе название, checkbox для отметки выполнения, а также дату создания, форматированную в виде, удобного для чтения.

3.2 Нефункциональные требования

1) Интуитивность и понятность интерфейса обеспечивает библиотека React с использованием material-UI графической библиотеки от Google. Эта библиотека даст приложению современный дизайн, а также адаптивность. К тому же благодаря этой библиотеке приложение сможет работать без перезагрузки страницы.

2) Безопасность. Так как приложение однопользовательское, его данные не сохраняются на сервере. Все личные данные пользователей сохраняются на их машинах, соответственно, данные можно заполучить только взломав непосредственно компьютер пользователя.

3) Надежность. Приложение одностраничное, однопользовательское, поэтому можно не беспокоиться за его работу. Единственное что может помешать его работе – это закрытие вкладки в браузере, но данные все равно сохранены, так что можно считать приложение надежным.

4) Валидация данных это очень важный пункт, т.к. защита от взлома и надежность – это все что важно. Нужно предусмотреть обработку ошибок пользователя. К примеру, это создание задачи с пустым полем. Необходимо обеспечить валидацию форм для предотвращения поломки в приложении.