В свободное время можете попробовать порешать гугля, задачки более похожи к реальным

1. Форма регистрации с валидацией полей:

Создайте HTML-форму(либо же можете почитать про Formik, React-Hook-Form и использовать данный вид формы) с полями для ввода имени, электронной почты и пароля.

Используйте JavaScript для проверки введенных данных: имя не должно быть пустым, электронная почта должна содержать символ "@", пароль должен содержать не менее 6 символов.(для валидации можете прочитать про yup validations и конечно же посерчить регулярки под это дело)

Выведите сообщения об ошибках или подтверждение успешной регистрации(для высплывающих нотификаций можно использовать npm пакет react-toastify можете почитать).

1. Простое приложение "Заметки":

Создайте интерфейс для добавления, удаления и редактирования заметок.

Реализуйте функциональность фильтрации заметок по ключевым словам

1. Работа с API(по желанию):

Используя публичное API (например, OpenWeatherMap), напишите функцию, которая по городу возвращает текущую погоду. Интегрируйте это в простой веб-интерфейс, где пользователь вводит название города, а затем отображается текущая погода.

1. Аутентификация с использованием локального хранилища(по желанию):

Реализуйте механизм аутентификации с использованием локального хранилища. Создайте форму входа, где пользователь вводит имя пользователя и пароль. Если данные верны, сохраните информацию об аутентификации в локальное хранилище, чтобы пользователь оставался в системе после перезагрузки страницы.

для Node.js тоже по желанию, так как в доскональности не помню вашу программу, но в любом случае если есть свободное время и захотите что-то посмотреть\сделать- ловите   
  
  
Простой веб-сервер:

Создайте простой веб-сервер с использованием Node.js и Express. Настройте несколько маршрутов (например, главная страница, страница о нас), а также обработку статических файлов (CSS, изображения).

Чтение и запись в файл:

Напишите программу, которая читает данные из текстового файла, модифицирует их (например, изменяет регистр букв) и записывает обратно в файл. Это может быть, например, простой логгер с возможностью анализа и модификации данных.

Работа с базой данных:

Подключите базу данных (например, MongoDB с использованием Mongoose) и создайте простую схему данных (например, пользователи). Реализуйте операции CRUD (Create, Read, Update, Delete) для этих данных.

Аутентификация с использованием Passport.js:

Интегрируйте библиотеку Passport.js для реализации простой системы аутентификации на основе локальной стратегии (имя пользователя и пароль). Создайте маршруты для входа и выхода из системы.

Работа с API стороннего сервиса:

Используя библиотеку Axios, напишите код для взаимодействия с API стороннего веб-сервиса (например, получение данных о погоде, новостях и т.д.) и возвращения этих данных через ваш веб-сервер.

Разработка RESTful API:

Создайте простой RESTful API с использованием Express. Реализуйте эндпоинты для создания, чтения, обновления и удаления ресурсов. Протестируйте ваше API, используя инструменты, такие как Postman.