****

The purpose of this document is to outline the setup of this project's GitHub organization, repositories and the uses for each.

## Table of Contents

[**Table of Contents**](#_sivvkfdgradb)2

[GitHub Organization](#_maxr3iw6l7tx) 3

[GitHub Repositories](#_q4vohtcwhgja) 5

[GitHub Commit Messages](#_9ux94bkkaa0d) 5

[Frontend](#_ebp4lwd9pm0l) 6

[Static-Frontend](#_ebp4lwd9pm0l) 8

[Documents](#_a41n606tizk3) 9

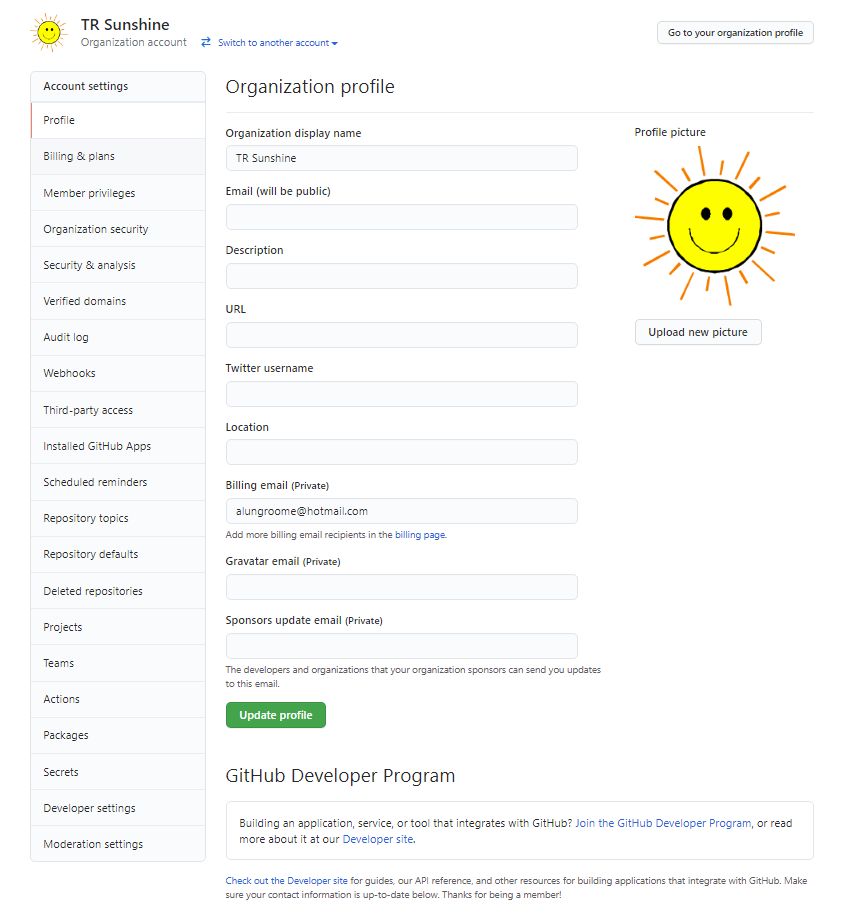
[Backend](#_rb4bs0lnk0vv) 10

## GitHub Organization

A GitHub organization consists of shared accounts where businesses and open-source projects can collaborate. Owners and administrators can manage member access to the organization's data and projects.

The GitHub Organization name for this project is ‘TR Sunshine’.

Setup

1. Create the organization for the project group named ‘TR\_Sunshine’.
2. Invite all members of the project team to the GitHub organization.
3. Assign all project members ‘Owner’ Access to the organization.

## GitHub Repositories

A GitHub repository contains all of your project’s files and each file’s revision history. You can use repositories to manage your work and collaborate with others. One of the most common uses for GitHub repositories is to version control source code.

### GitHub Commit Messages

The below naming convention should be used when committing any change to any repository in this project:

Type of change

* FEAT: The new feature you're adding to a particular application
* FIX: A bug fix
* STYLE: Feature and updates related to styling
* REFACTOR: Refactoring a specific section of the codebase
* TEST: Everything related to testing
* DOCS: Everything related to documentation

Usage

1. The first word of the commit should be the keyword from the ‘Type of change’ listed above with capitalization of each character.
2. Followed by a semicolon and space character.
3. Followed by a brief description of the change where the first character of the sentence is capitalized.
4. Total length of the commit message is limited to 50 chars.

Example

For fixing a bug in code, the commit message would be similar to below:

FIX: Null Exception in Function GetWeather

### **Frontend**

As part of this project the Frontend repository will be used to version control the source code for the projects Frontend which is developed using React and Bootstrap.

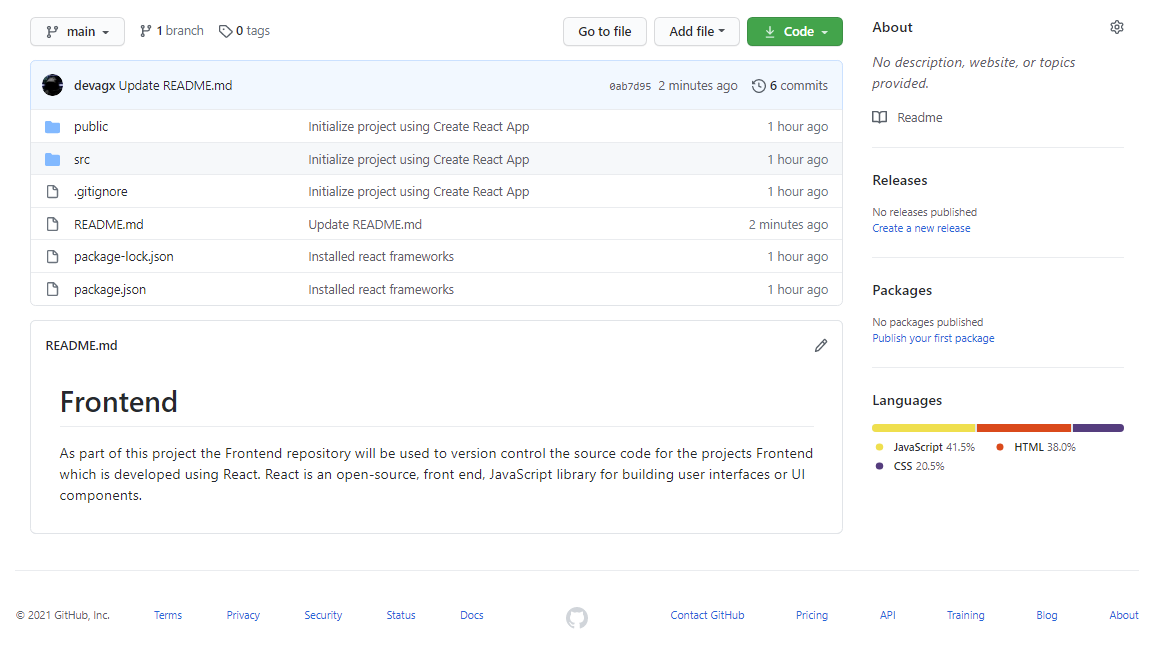
React is an open-source, front end, JavaScript library for building user interfaces or UI components.

Bootstrap includes a responsive, mobile first fluid grid system that appropriately scales up to 12 columns as the device or viewport size increases.

Initial Setup

The below outlines the steps needed for the initial setup. These steps should not be performed in an attempt to clone a repository.

1. Create a new public repository named ‘Frontend’
2. Create Initial react code base via installation under under main frontend repository
   1. npx create-react-app tr\_sunshine\_react\_application
3. Link GitHub Repository to local code
   1. Navigate to project directory
      1. cd tr\_sunshine\_react\_application
   2. Link GitHub repository to local code
      1. git remote add origin https://github.com/TR-SUNSHINE/Frontend.git
      2. git branch -M main
      3. git push -u origin main
4. Install additional frameworks and components
   1. React Routers
      1. npm install react-router-dom
   2. Material UI
      1. npm install @material-ui/core
   3. React Bootstrap
      1. npm install react-bootstrap bootstrap
5. Sync to GitHub
   1. git add .
   2. git push -u origin main



Steps to clone the repository

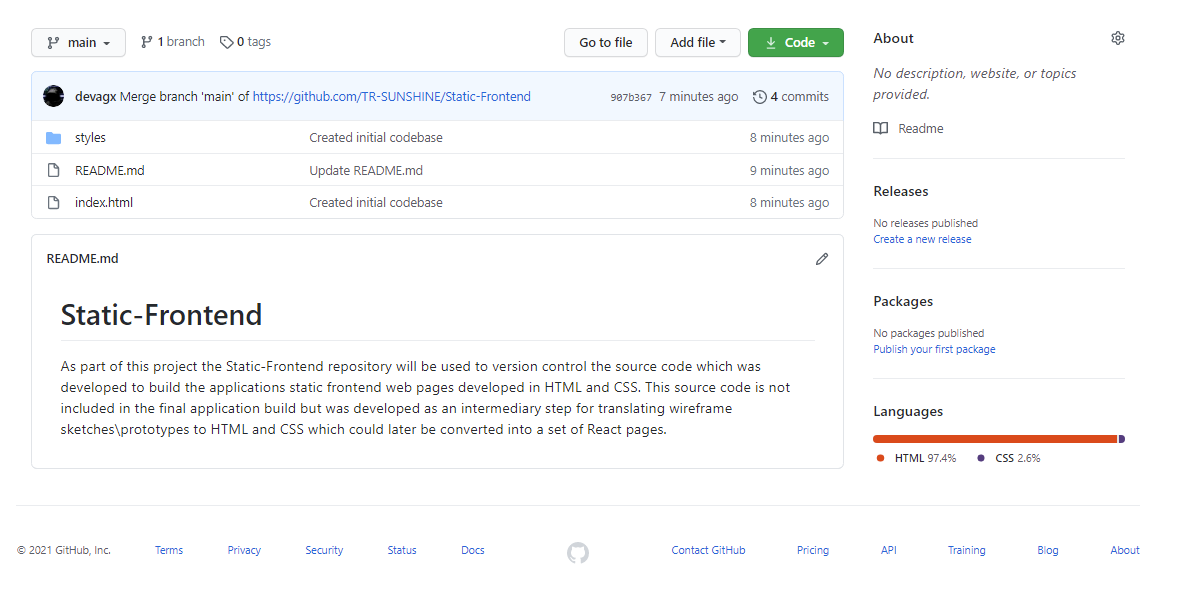
1. Navigate to working directory
2. git clone <https://github.com/TR-SUNSHINE/Frontend.git>
3. Install additional frameworks and components
   1. React Routers
      1. npm install react-router-dom
   2. Material UI
      1. npm install @material-ui/core
   3. React Bootstrap
      1. npm install react-bootstrap bootstrap

### **Static-Frontend**

As part of this project the Static-Frontend repository will be used to version control the source code which was developed to build the applications static frontend web pages developed in HTML and CSS. This source code is not included in the final application build but was developed as an intermediary step for translating wireframe sketches\prototypes to HTML and CSS which could later be converted into a set of React pages.

Initial Setup

1. Create a new public repository named ‘Static-Frontend’.
2. Create template HTML and CSS code files.
3. Link Github repository to local code
   1. echo "# Static-Frontend" >> README.md
   2. git init
   3. git add README.md
   4. git commit -m "first commit"
   5. git branch -M main
   6. git remote add origin https://github.com/TR-SUNSHINE/Static-Frontend.git
   7. git push -u origin main
   8. git add .
   9. git push origin main



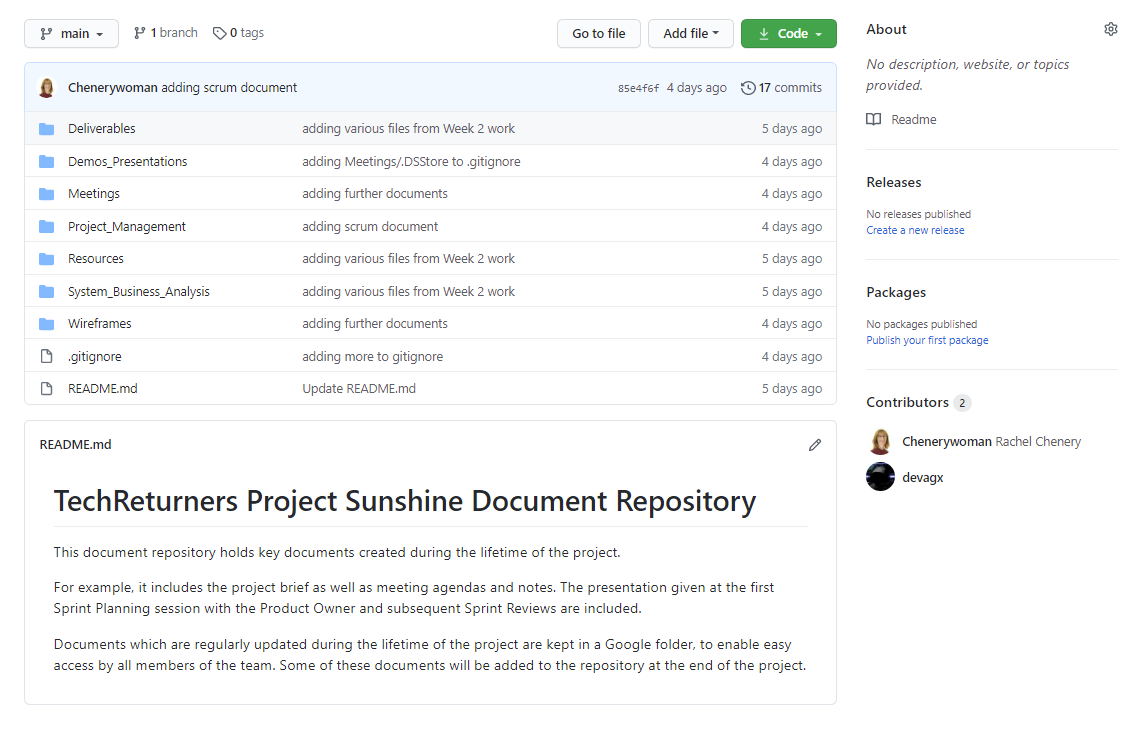
Steps to clone the repository

1. Navigate to working directory
2. git clone https://github.com/TR-SUNSHINE/Static-Frontend.git

### 

### **Documents**

As part of this project the Documents repository will be used to publicly store documentation which can be shared with TechReturners and associated businesses.



Steps to clone the repository

1. Navigate to working directory
2. git clone https://github.com/TR-SUNSHINE/Documents.git

### **Backend**

TBD.