

## Using JavaScript created Covid-19 Employee Detection Website

### Azure Service used: Static Web app

---

#### Problem statement –

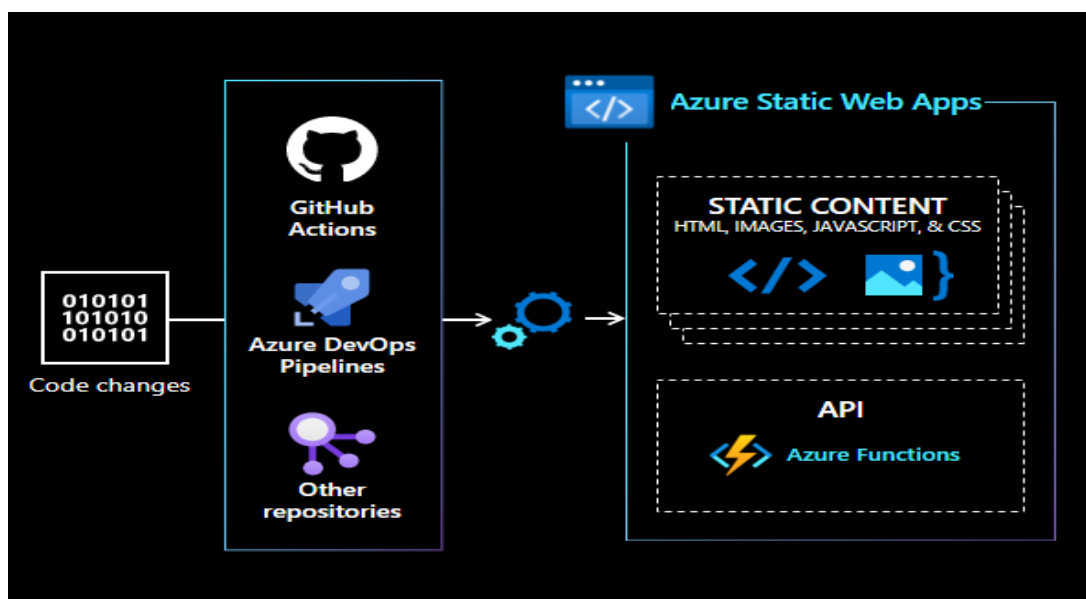
The start of the COVID-19 pandemic has given rise to digitalisation of data. Big Corporations have already established systems to monitor employee health conditions and analyze covid related employee data. Even small scale companies attempt to collect data and keep track of the health state of their employees. However, this is a complete physical process as they need to physically collect the data and later collectively analyze it. This makes the process of analyzing and monitoring data cumbersome and time-consuming.

Therefore, there is a need for an online system which can help small organizations to collect the data for further analysis. An easy to use system which can collect, store and further analyze the data for the smaller companies to better understand the health state of the organization.

#### Solution -

We have developed a system which helps the organization to gather health related data at a common place/database which then is used by us to convert it into some valuable information. This information shows the health state of the employee, if the employee is showing any ill symptoms, etc.

The administrators can also view the whole organization's overall state by viewing the classifications (which we call zones). The zones are Red, Yellow and green which optimally gives an overall perspective of the employees. The way this would work technically is as follows. Every employee and administrator will have a unique account on our website. After logging in employees will be asked to provide their health parameters on a daily basis through a form. Thus, this system converts the whole traditional manual process of collecting and analyzing data in the organization into effortless online mode. It also provides a viable solution to small scale industries/startups which cannot afford to go for high end expensive systems in their organization.



## Azure Static Web Apps

Azure Static Web Apps is a service that automatically builds and deploys full stack web apps to Azure from a code repository.

The workflow of Azure Static Web Apps is tailored to a developer's daily workflow. Apps are built and deployed based off of code changes.

When you create an Azure Static Web Apps resource, Azure interacts directly with GitHub or Azure DevOps, to monitor a branch of your choice. Every time you push commits or accept pull requests into the watched branch, a build automatically runs and your app and API deploys to Azure.

Static web apps are commonly built using libraries and web frameworks like Angular, React, Svelte, Vue, or Blazor where server-side rendering isn't required. These apps include HTML, CSS, JavaScript, and image assets that make up the application. With a traditional web server, these assets are served from a single server alongside any required API endpoints.

### Key Features:

- Web hosting for static content like HTML, CSS, JavaScript, and images.
- Globally distributed static content, putting content closer to your users.
- Free SSL certificates, which are automatically renewed.
- Custom domains to provide branded customizations to your app.
- CLI support through the Azure CLI to create cloud resources, and via the Azure Static Web App CLI for local development.

### Implementation of Azure Static web app :

You need the following software/ Azure Account, please find details below:-

Azure Account :- get an azure account by clicking on the following link

<https://azure.microsoft.com/en-us/free/>

There is free credit for students and 200 USD credit if you want to get started with Azure

<https://azure.microsoft.com/en-us/free/students/>

Visual Studio: <https://visualstudio.microsoft.com/downloads/>

# Steps to connecting Github pages to Azure Static Web App

## Step 1 : Deploy your project on your github pages

### GitHub Pages

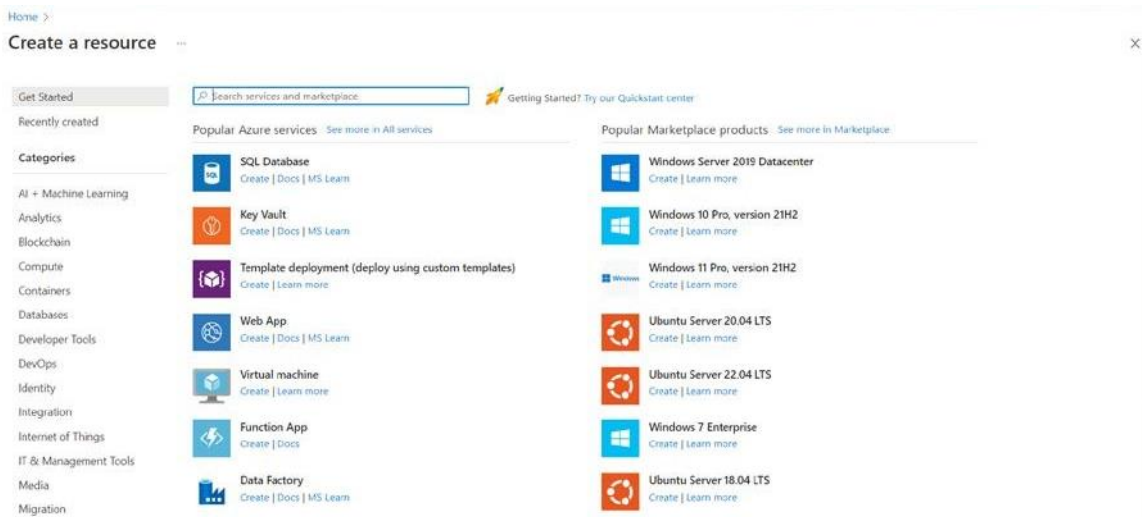
GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.

Your site is live at <https://octo-org.github.io/octo-repo/>  
Last deployed by  octocat 4 minutes ago

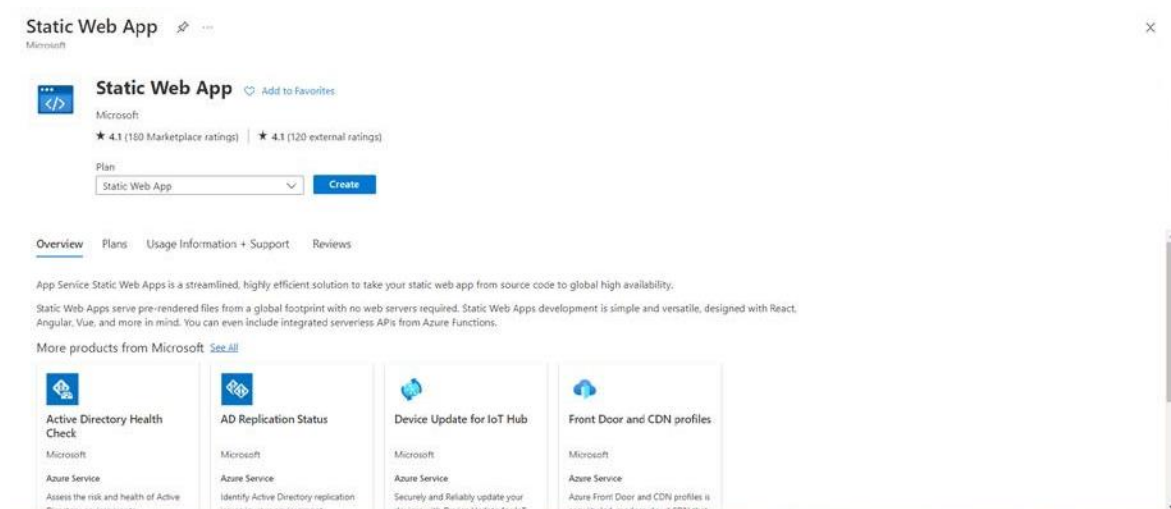
[Visit site](#)

...

**Step 2: Create an Azure Account and log in to the Azure Portal. After Azure account is created , Click on Create resource.**



**Step 3 : Search for Azure Static Web app and click on Create button.**



**Step 4: Fill up all the details Properly. Select the subscription which you have got while creating your free azure account. Select the resource group which you have created earlier or create a new resource group. Then Select Free plan. Then For deployment Select Github and do sign-in for connecting Github pages to Static Web app.**

## Create Static Web App ...

**Basics** Tags Review + create

App Service Static Web Apps is a streamlined, highly efficient solution to take your static app from source code to global high availability. Pre-rendered content is distributed globally with no web servers required. [Learn more](#) ↗

### Project Details

Select a subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription \* ⓘ

Azure for Students



Resource Group \* ⓘ

Alia

[Create new](#)

### Static Web App details

Name \*

covid19

### Hosting plan

The hosting plan dictates your bandwidth, custom domain, storage, and other available features. [Compare plans](#)

Plan type

Free: For hobby or personal projects

Standard: For general purpose production apps

### Azure Functions and staging details

Region for Azure Functions API and staging environments \*

Central US

### Deployment details

Source

GitHub  Azure DevOps  Other

**Step 5 : After signing in, select proper organization of the github , repository of the project, and branch of the website which want to connect with static web app.**

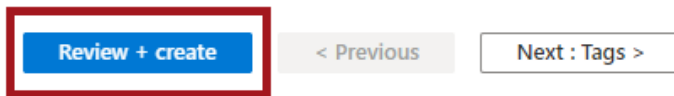
**i** If you can't find an organization or repository, you might need to enable additional permissions on GitHub. ×

Organization \*

Repository \*

Branch \*

**Step 6 : Select Review + create**



**Step 7 : After Reviewing the entered details correctly, click on Create then it will started deployment . Then after successful deployment , you will get the deployment URL of theproject.**

**Microsoft.Web-StaticApp-Portal-35cd0333-8dcf | Overview** 🔍 ⋮

Deployment

⏪ 🗑️ Delete ⌛ Cancel 🔄 Redeploy ⬇️ Download 🔄 Refresh

- Overview**
- Inputs
- Outputs
- Template

**✔️ Your deployment is complete**

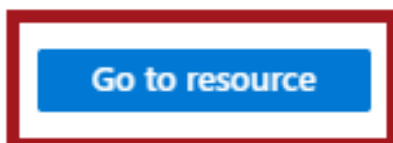
**📄** Deployment name: Microsoft.Web-StaticApp-Portal-35... Start time: 1/12/2023, 9:03:48 PM  
Subscription: Azure for Students Correlation ID: fc579a7f-e73d-4d1b-9333-23a5fe824  
Resource group: Alia

∨ Deployment details

∧ Next steps

[Go to resource](#)

**Step 8 : Click on go to resource it will take you to the static web app where you will get the link of the static web app website.**



## Step 9 : Now, Your Static App is successfully deployed. Click on the URL to go the website.

The screenshot shows the Azure portal interface for a Static Web App. The left sidebar contains navigation options: Overview, Access control (IAM), Tags, Diagnose and solve problems, Settings, Configuration, Application Insights, Custom domains, APIs, Environments, Role management, Identity, Enterprise-grade edge, Hosting Plan, Private endpoints, and Locks. The main content area displays the 'Essentials' section with the following details:

- Resource group (move) : [Alia](#)
- Subscription (move) : [Azure for Students](#)
- Subscription ID : 7f36600e-83d4-4d3e-84a0-f4e71ddf526a
- Location : Global
- Skus : Free
- Tags (edit) : [Click here to add tags](#)
- URL : <https://lively-dune-04c68b510.2.azurestaticapps.net>
- Source : [main \(GitHub\)](#)
- Deployment history : [GitHub Action runs](#)
- Edit workflow : [azure-static-web-apps-lively-dune-04c68b510.yml](#)

Below the Essentials section, there are two empty line charts labeled 'Requests' and 'Data out', both showing a y-axis from 30 to 100 and a few data points near the bottom.

The screenshot shows a web browser displaying the 'COVID 19 DETECTION' application. The browser's address bar shows the URL [lively-dune-04c68b510.2.azurestaticapps.net](https://lively-dune-04c68b510.2.azurestaticapps.net). The application has a green sidebar with navigation links: Dashboard, Profile, Table, Login, and Register. The main content area is titled 'Dashboard' and features the following components:

- EMPLOYEE HEAD COUNT**: 180
- Marketing**: 40
- Sales**: 30
- Testing**: 40
- Finance**: 30
- Development**: 40
- Production**: 40
- Employee Zone Distribution**: A donut chart showing the distribution of employees across three zones: Red Zone (approximately 50%), Green Zone (approximately 30%), and Yellow Zone (approximately 20%).
- Top Symptoms**: A horizontal bar chart showing the prevalence of symptoms: Fatigue at 30% and Dry Cough at 10%.

## Few Screenshots of the Web App:

**HEALTHIFY**

Abhiraj Arora

### Profile

Green Zone

**Personal Settings**  
#ID 198767890

**Software Developer**

Username: Abhiraj.Arora | Email Address: AbhirajArora@company.com

First Name: Abhiraj | Last Name: Arora

**Personal Info**

Department: Development

Phone: +91 9999999999 | Country: India

**Health Parameters**

Fatigue	30%
Dry Cough	10%
Cold	5%
Fever	3%

**HEALTHIFY**

### Update your daily Health Status

Do you have any Chest Pain?  
 Yes  No

Feeling Fatigued?  
 Yes  No

Temperature

Is there any Short Breath?  
 Yes  No

Do you have any Cough?  
 Yes  No

Is there any Tiredness?  
 Yes  No

Body Ache/Pain?  
 Yes  No

## **Conclusions and further work**

- Thus, we were able to design and integrate the frontend and backend for our Website.
- The Website has a seamless user experience with automatic updation on the server side.
- Every change reflects for the employee as well as the admin and the departments get sorted as per zones.
- For Future Scope, we utilise the website to monitor other health related complexities of the employees.

## **Azure Static Web App URL:**

<https://lively-dune-04c68b510.2.azurestaticapps.net/profile.html>

## **Challenges Faced:**

- Problem was faced during the deployment of azure static web app. Sometimes After entering the details in the creation panel the options were not coming fast like for selecting github repo
- It was taking more time for deployment of Static Web app
- After Performing some changes and pushing to the github, it takes more time to get the changes updated in the Azure Static Web app,
- There was connection lost issue sometimes

## **Business Benefits:**

- Static websites are quick to set up, lightning fast and can be easily maintained
- If you plan to just put up a few pages that won't change very often, the maintenance fees are minimal. A static website can be a good solution for those who don't want to pay for web hosting or a content management system.
- Static websites are resilient to hacking attacks. This is because they do not contain any databases, and therefore cannot be subject to database injection attacks.