This web application consists of a single HTML page, making this a static website.

Design phase

Choosing between on-premises and cloud

I have considered three main factors to choose between on premises and cloud. These factors are cost, security & performance.

Cost

A crucial point in cloud computing vs on-premise cost comparison is an initial investment in hardware and infrastructure.
Costs can be high for on-premise with ongoing costs of maintenance, storage space, power consumption and dedicated IT support. Infrastructure also requires regular investment to stay up to date.
With cloud computing the Capital expenditure (CAPEX) is practically non-existent, as no hardware is needed, nor any physical infrastructure to hold servers. No need of IT staff to maintain the network. Maintenance or upkeep costs are kept to a minimum as there is no physical hardware or infrastructure to maintain.
Also cloud services have 'pay as you go' mechanism which helps us scale up and down to suit market demands, however for on-premise since the hardware is fixed it can be underutilized wasting investment for unused hardware.

Security

Highly-secure data encryption is essential for any online business and with cloud computing, it's all included in the package.
One might think on-premises would be the safer option for protecting sensitive data, because critical data is kept in-house and no third-party has access to your information but, almost all the massive data breaches we've had over the last decade have been from traditional on-premise IT infrastructures.

Performance

Cloud computing is generally used on 'as-needed' basis, users only pay for what they use, the cloud load balancer automatically handles increase in network traffic and high demands. For on-premise, since the hardware is fixed it can be overutilized causing performance issues.

Decision

Based on these factors I have decided to go with cloud computing services.

Which cloud computing service ?

We have a plethora of cloud service providers in the market. For ease of comparison I have chosen **free** subscription plans of AWS, MS Azure and GCP into consideration.

		AWS	MS Azure	GCP
	App name	S3	Static web apps	Cloud storage
Cost	Subscription	Free	Free	Free
Storage and compute	Max storage	5GB	.5GB per app	5GB
	Freebies validity	12 Months	Always	Always
Security	Protocol	HTTP	HTTPS	HTTPS
Performance	Regions	All nodal points	All nodal points	US region only

Decision

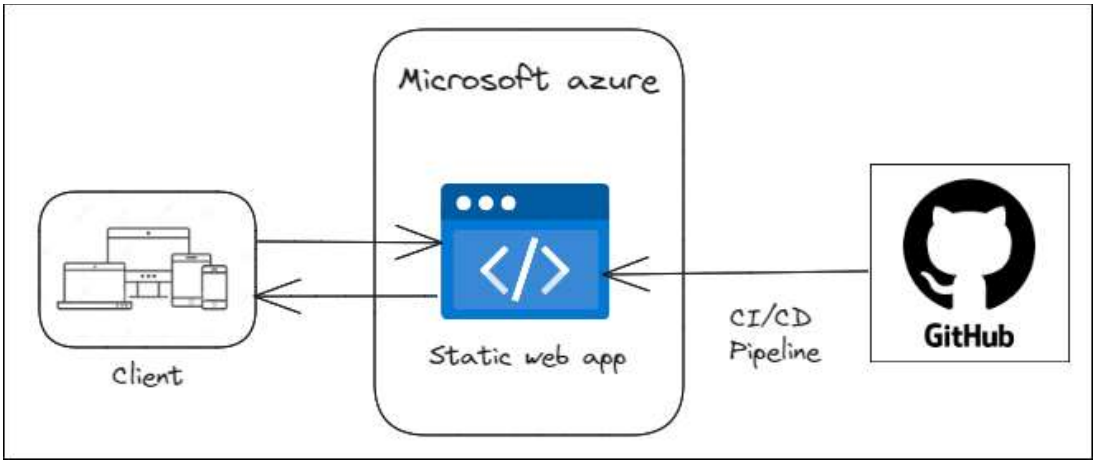
Among these three, MS Azure is the best fit because it provides **better security** than AWS by allowing HTTPS transfers and it provides **better performance** than GCP with nearest nodal points in Chennai and Pune. Since this is a static web app with little storage, .5GB is sufficient to store the web resources. In addition to above Azure Static web app implements **HTTPS enforcer** by default.

Code repository and source control

I have chosen GitHub as my code repository. This is an open source cloud repository(SaaS) and provides the flexibility to maintain the repository as public or private. GitHub also provides wide range of modern features like collaborative coding, automation and CI/CD, project management etc. all these feature help in maintaining and delivering quickly. **Git hub automatically creates a CI/CD pipeline for azure static web app deployment.**

Architecture diagram

Below diagram depicts the architecture of the application

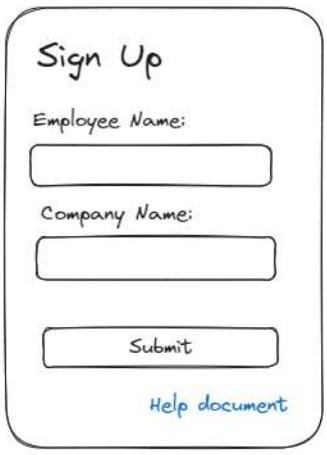


UI design

The UI consists of text input elements, their labels and a submit button. The UI must support PC and mobile devices

Wireframe

Below is a mock diagram of how the screen will look



Development phase

The requirements and strict security needs of the application are considered during development process. We will discuss them in detail below

Requirement

The website must support PC and mobile devices.
Also it is one of steps for search engine optimization to optimize 'title' tags, using 'meta' description and 'meta' keywords. I have included these changes in the project to support SEO.

PC and mobile screen support

For the website to support varying screen sizes I have used <meta> elements 'viewport' attribute

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

Search engine optimization support

SEO is a major aspect for the **marketing of an application**. This helps the website to rank higher in the search results of a search engine, and helps boost the **income**.
'description' attribute of the <meta> element provides the description of the page that search engines can pick up to show during the result

```
<meta name="description" content="Sample website">
```

'keywords' attribute of the <meta> element provides a list of comma-separated keywords relevant to the page, which are used by the search engine to categorize the page

```
<meta name="keywords" content="TTEC, Sample Website">
```

Web security

Web application security is critical to protect web servers and web applications from attack by internet based threats.

HTTPS protocol

Hypertext Transfer Protocol Secure is an extension of the Hypertext Transfer Protocol. It uses encryption for secure communication over a computer network, and is widely used on the Internet. In HTTPS, the communication protocol is encrypted using Transport Layer Security(TLS). Azure Static web app automatically redirects HTTP requests to HTTPS, this action is commonly known as **HTTPS enforcer**.

XSS (Cross site scripting) attacks (or) HTML injection

It is a security vulnerability that allows an attacker to inject HTML code into web pages that are viewed by other users. This threat can be prevented by using 'http-equiv' attribute of <meta> element.

```
<meta http-equiv="content-security-policy" content="default-src 'self'; style-src 'self' 'unsafe-inline';">
```

Reverse tab nabbing

Reverse tab nabbing is an attack where a page linked from the target page can rewrite that page. This threat can be avoided by using 'rel' attribute of anchor(<a>) element.

```
<a href="./documents/HelpDocument.pdf" rel="noopener noreferrer" target="_blank">Help document</a>
```

Launch phase

Web hosting

As discussed earlier MS Azure is our cloud service provider which provides computing services through **'Static web app'**. This provides **free** hosting with HTTPS by creating necessary SSL certificates along with free domain name, available across all nodal points (with nearest locations in Chennai and Pune).

Deployment

Deployment is the process by which all the project related artifacts are made available for use.

CI/CD pipeline

GitHub creates a '.yml' file automatically to support continuous deployment with MS Azure Static web app. With this, any changes done to 'main' branch gets deployed automatically to the host.

Documentation

The documentation helps keep track of all the aspects of an application, and it improves the quality of the software product.

There will be a help document available for this app which can be accessed from the UI. This document will be stored and accessed from the code repository.

Best practices followed

1. Included HTML lint and CSS lint in the project for applying rules
2. Included prettier in the project for applying beautification
3. SEO optimization support
4. Handled 400(Bad Request) & 404(Page not found) HTTP errors by using 'staticwebapp.config.json' configuration file
5. HTTPS enforcer
6. Continuous Deployment(CD) pipeline to automatically deploy the code to application host
7. Updated 'readme.md' file in the code repository with relevant steps

Other possible best practices

Implementing Continuous integration(CI) with automated pipelines to verify (using lint) and beautify(using prettier) the code files in the project.

GitHub repository link: https://github.com/SivaChinchalapu/SampleWebsite/