**PROJECT1:PERSONAL BLOG ON IBM CLOUD STATIC WEB APPS**

**Phase 1**

**Problem Definition and Design Thinking**

**Problem Definition**

A personal blog allows you to share your creativity and style with an online community. The great thing about starting a personal blog is that you can be a beginner.

**Design Thinking:**

1. ***Empathize***:  Empathy is to set aside our own thoughts and assumptions about the world in order to gain insight into the users and their needs.
2. ***Define***: During the Define stage, you are expected to write down the problem statement**.**
3. ***Ideate:*** In this stage, designers are ready to start generating ideas. You and your team members can now start to “think outside the box” to find out various possible solutions to problems.
4. ***Prototype:*** Now it’s time to convert ideas into working prototypes that cost less and take less time. his is done so that they can be evaluated and the problems associated with them can be analyzed.
5. **Test**:This is the last stage of the design thinking process but as it is an iterative process, the results generated during the testing phase are often used to redefine problems. Alterations and refinements are carried out in this phase in order to rule out problems.

PHASE 2-INNOVATION

The innovation phase is the critical step in which we transform our design thinking ideas into a tangible cloud based static web apps. This document outlines the comprehensive steps that will be taken to put design concept into practice.

**DEFINITION USING FLASK**

A personal blog is a website or online platform where an individual or a group of individuals share their thoughts, experiences, knowledge, interests, and perspectives on a wide range of topics. It is a digital diary or journal that provides a space for people to express themselves,connect with an audience, and communicate their ideas to the world.

Key characteristics of a personal blog include:

1. **Authorship:** Personal blogs are typically authored by one or a few individuals who share their personal experiences, insights, and expertise.

**2. Diverse Content:** Personal bloggers can cover a wide range of topics, including travel, food, fashion, technology, lifestyle, personal development, and more.

3**. Informal Tone:** Personal blogs often have a conversational and informal writing style, making them relatable and engaging for readers.

4. **Personal Perspective**: Bloggers inject their unique voice and perspective into their content, sharing their thoughts, opinions, and personal anecdotes.

5. **User Interaction:** Blogs often allow for reader interaction through comments, allowing readers to engage with the author and share their own thoughts.

6**. Regular Updates**: Successful personal blogs are frequently updated with new content, helping to build and maintain an audience.

7. **Independence:** While some personal blogs may have affiliations with brands or advertisers, they are typically independently operated and not tied to traditional media organizations.

Personal blogs can serve various purposes, from simply sharing personal experiences and hobbies to promoting products, building personal brands, or even generating income through advertising, sponsored posts, or affiliate marketing. They provide a platform for individuals to express themselves, connect with like-minded individuals, and potentially reach a global audience with their ideas and stories

Integrating social media sharing buttons, interactive maps, and comment sections into your Flask web application can significantly enhance user engagement and interaction. Here's a detailed guide on how to implement each of these features:

1. **Social Media Sharing Buttons:**

Social media sharing buttons allow users to easily share your blog posts or content on their preferred social platforms. You can implement these buttons using third-party JavaScript libraries or custom HTML and CSS.

**Step 1: Get Shareable Content:**

Make sure your blog posts or content have shareable URLs. Each post should have a unique URL to share.

**Step 2: Choose a Social Sharing Library:**

There are various libraries and services to choose from, such as AddThis, ShareThis, or simple HTML/CSS solutions.

**Step 3: Include JavaScript Libraries:**

If you choose a JavaScript-based solution, include the necessary JavaScript library in your Flask template. For example, if using AddThis, you would include their script in your HTML:

Code:

<script type=”text/javasrcipt”

Src=”//s7.addthis.com/js/300/addthis\_widget.js#pubid=YOUR\_PUB\_ID”

></script>

**Step 4: Create Share Buttons:**

In your template, create HTML elements for social media sharing buttons. With AddThis, you can use code like this:

Code:<div class=”addthis\_inline\_share\_tool box”><\div>

1. **Interactive Maps:**

Interactive maps can add a dynamic element to your Flask application, allowing users to explore locations, events, or data visually.

**Step 1: Choose a Mapping Service:**

Select a mapping service, such as Google Maps, Mapbox, or Leaflet, to create and display your interactive maps.

**Step 2: Get an API Key:**

For services like Google Maps, you'll need an API key, which you can obtain by signing up for their developer program.

**Step 3: Include JavaScript Librarie**

In your Flask template, include the necessary JavaScript libraries and CSS stylesheets for your chosen mapping service.

**Step 4: Add a Map Container:**

Create an HTML container element where your map will be displayed.

Code:

<div id=”map”style=”width:100%;height:400px;”></div>

**Step 5: Initialize the Map:**

Use JavaScript to initialize the map, set markers, and define interactivity. For example, with Leaflet:

Code:

<script>

Var map=l.map(’map’).setView([51.505,-0.09],13);

L.tileLayer(‘https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png’,{

maxZoom:19,

}).addTo(map);

L.marker([51.5,-0.09]).addTo(map)

.bindpopup(‘A sample location’)

.openpopup();

</script>

1. **Comment Sections:**

Adding a comment section allows users to engage with your content by leaving comments and feedback. You can use third-party services or create a custom solution.

**Step 1: Choose a Comment System:**

You can use services like Disqus, Facebook Comments, or implement a custom comment system using Flask and a database.

**Step 2: Include the Comment System:**

For services like Disqus or Facebook Comments, you'll need to include their JavaScript code on your page.

**Step 3: Create a Database (if custom):**

If you're implementing a custom comment system, set up a database to store and retrieve comments.

**Step 4: Create a Comment Form:**

Design and implement an HTML comment form where users can input their comments.

**Step 5: Process and Display Comments:**

Use Flask to process submitted comments, store them in the database, and display them on the page.

**CONCLUSION:**

Integrating these features will enhance user engagement and interaction on your Flask web application, making it more appealing and interactive for your visitors

**PHASE-3**

**In this part you will begin building your project. Start by designing and developing the static travel blog website. Design the website layout using HTML and CSS. Create engaging content with captivating photos and travel stories.**

**Step1:planning and structure**

* Home
* Blog posts
* About
* Contact us

**Step 2: Create HTML Structure**

Create a simple website for using HTML code

<!DOCTYPE html>

<html>

<head>

<title>My Travel Blog</title>

<link rel="stylesheet" type="text/css" href="style.css">

</head>

<body>

<header>

<h1>My Travel Blog</h1>

<nav>

<ul>

<li><a href="index.html">Home</a></li>

<li><a href="blog.html">Blog</a></li>

<li><a href="about.html">About</a></li>

<li><a href="contact.html">Contact</a></li>

</ul>

</nav>

</header>

<main>

<!-- Content goes here -->

</main>

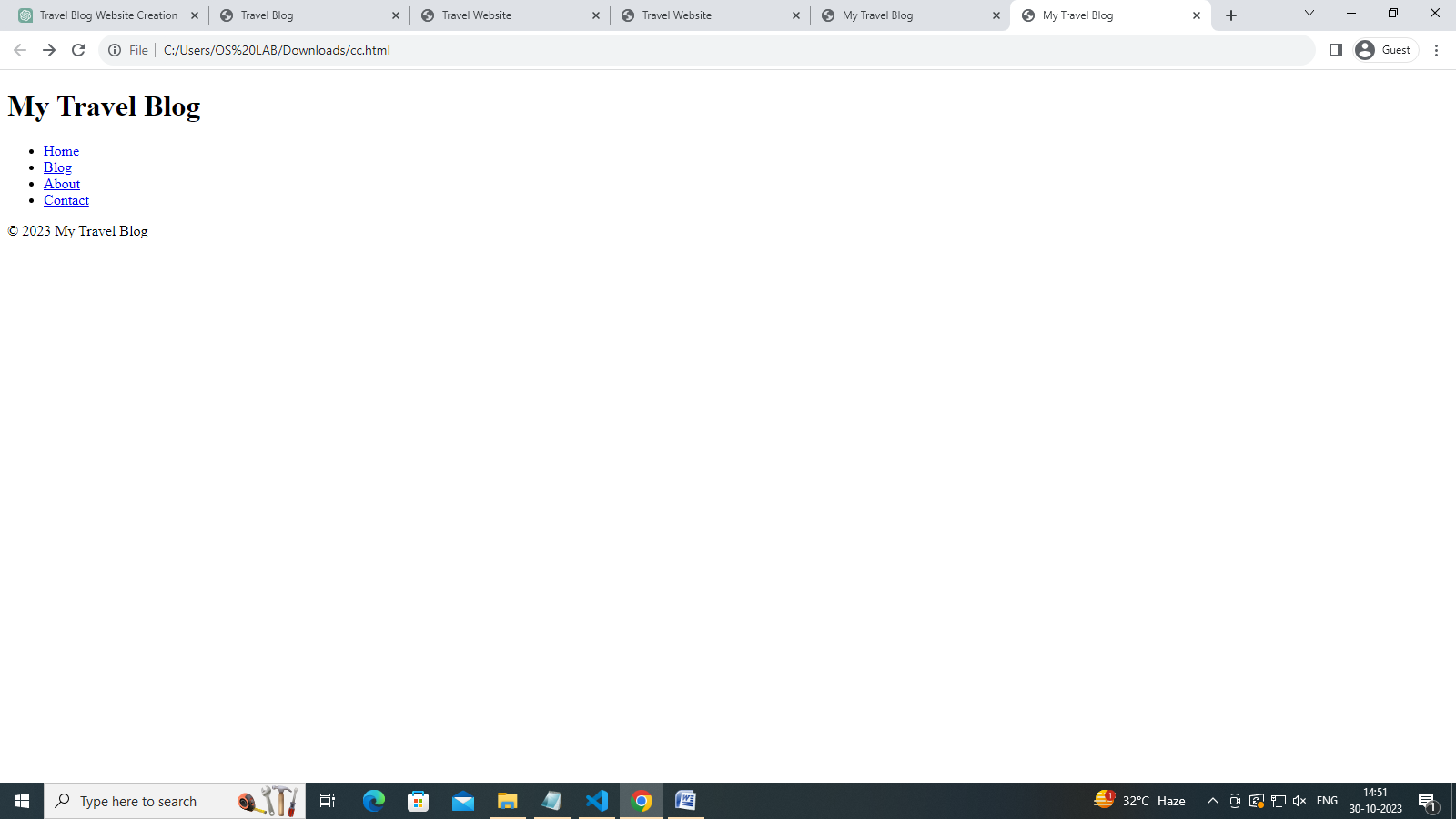
<footer>

&copy; 2023 My Travel Blog

</footer>

</body>

</html>



**Step3: add content**

Add content to our website

<!-- ... (previous HTML code) ... -->

<section class="destinations">

<h2>Popular Destinations</h2>

<div class="destination-card">

<img src="paris.jpg" alt="Paris, France">

<h3>Paris, France</h3>

<p>Experience the romance of the City of Love. Visit iconic landmarks like the Eiffel Tower and Louvre Museum.</p>

</div>

<div class="destination-card">

<img src="tokyo.jpg" alt="Tokyo, Japan">

<h3>Tokyo, Japan</h3>

<p>Discover the blend of tradition and modernity in Tokyo. Explore ancient temples and bustling neighborhoods.</p>

</div>

<!-- Add more destination cards here with images, titles, and descriptions -->

</section>

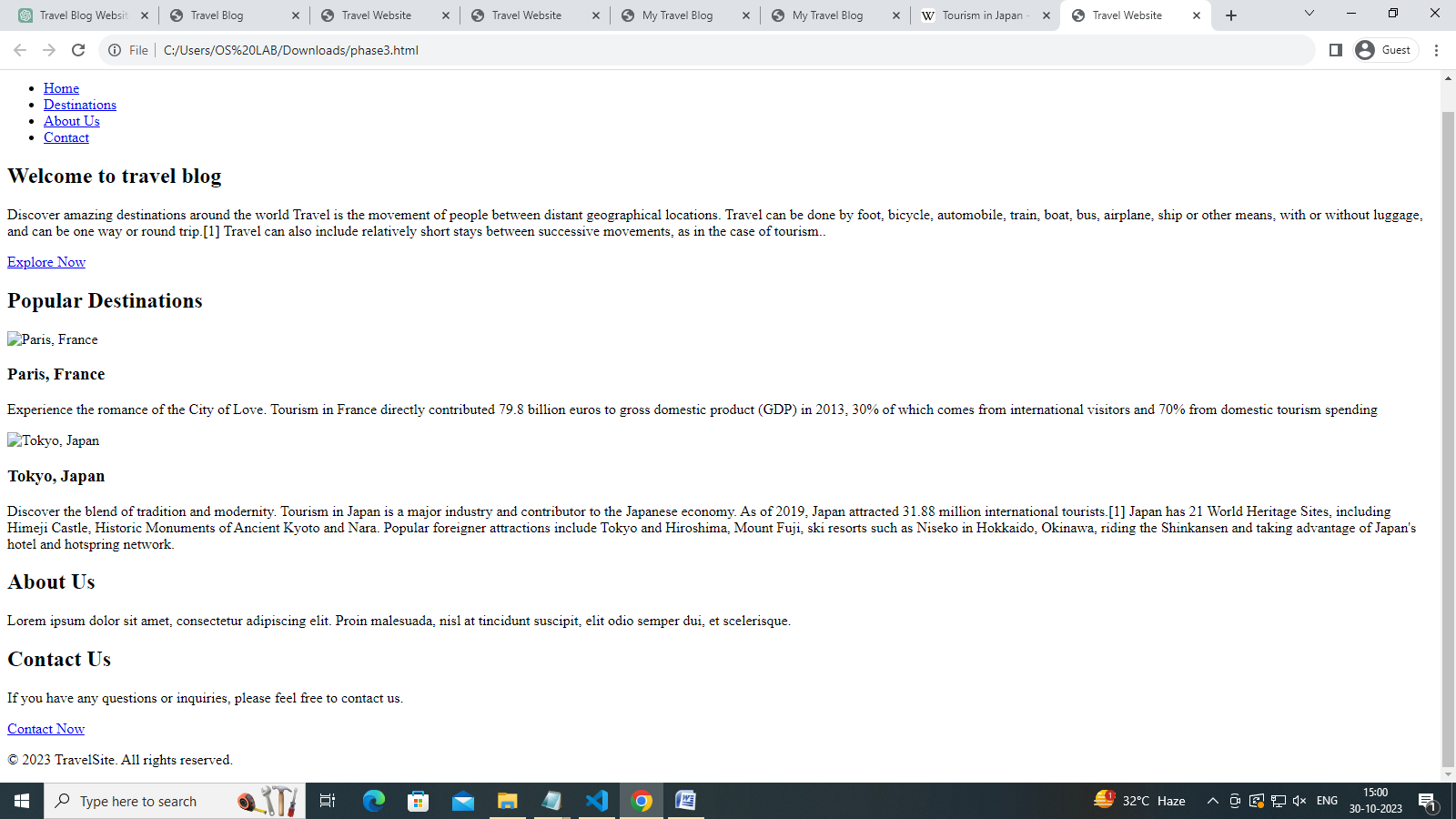
<section class="about">

<h2>About Us</h2>

<p>Welcome to TravelSite, your ultimate travel destination guide. We are passionate about helping you explore the world's most amazing places. Our mission is to provide you with the best travel experiences, tips, and recommendations.</p>

<p>With years of travel expertise, our team is dedicated to curating the perfect itineraries and travel information to make your dream vacation a reality. We aim to inspire your wanderlust and assist you in planning unforgettable journeys.</p>

</section>



**Step 5: Link Pages**

In the navigation menu in the header, make sure to link to the corresponding HTML pages. For example:

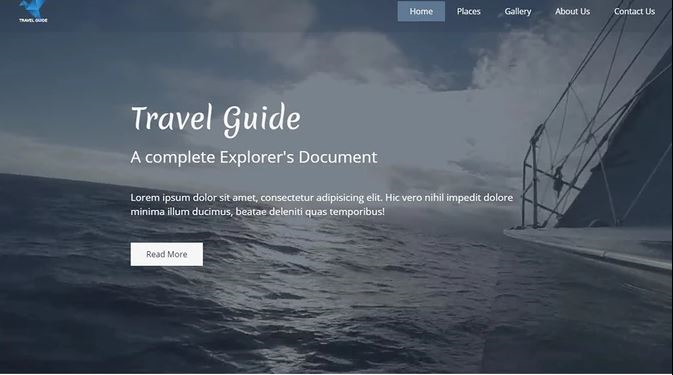
html

Copy code

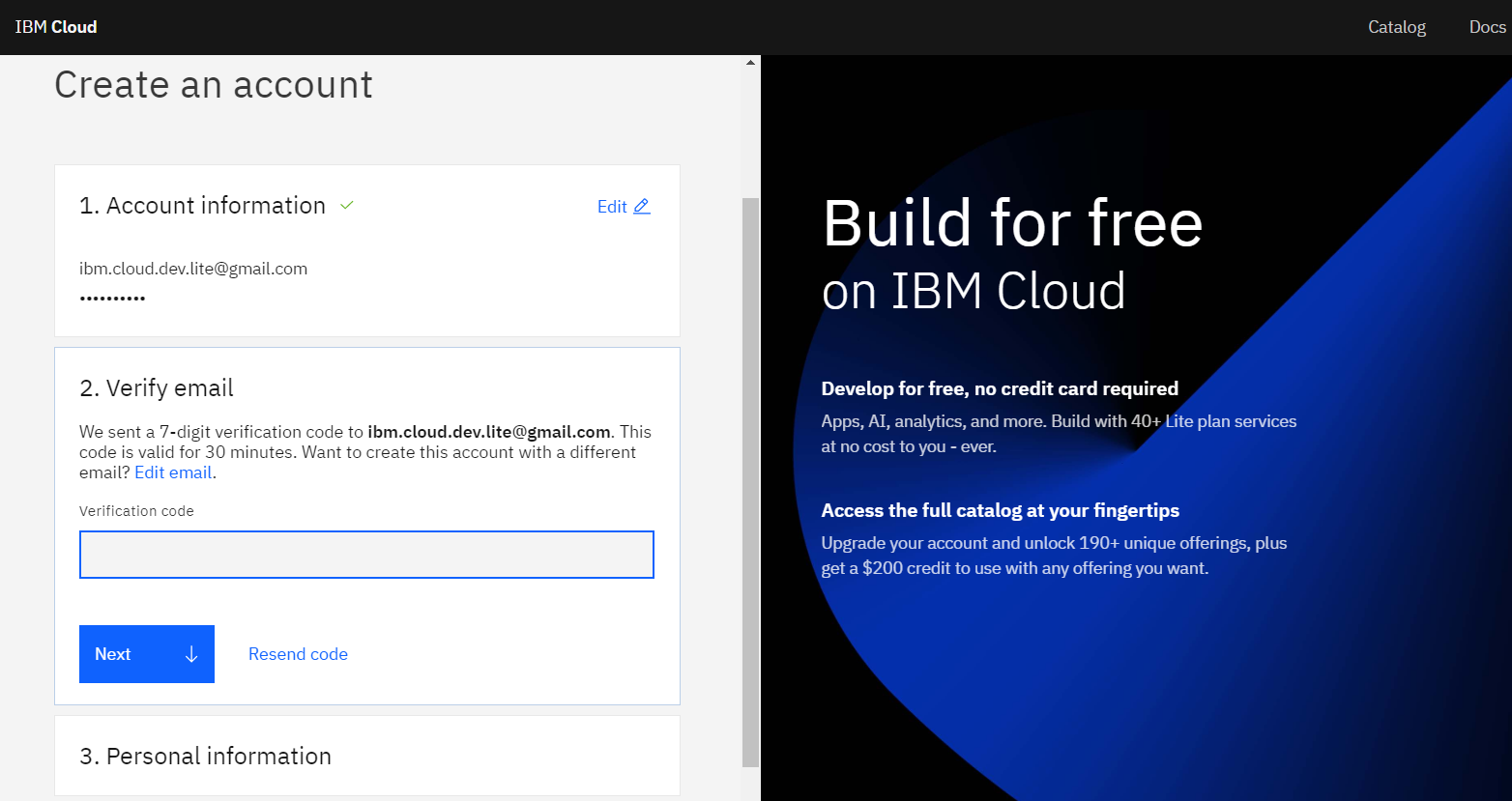
<li><a href="index.html">Home</a></li>

<li><a href="blog.html">Blog</a></li>

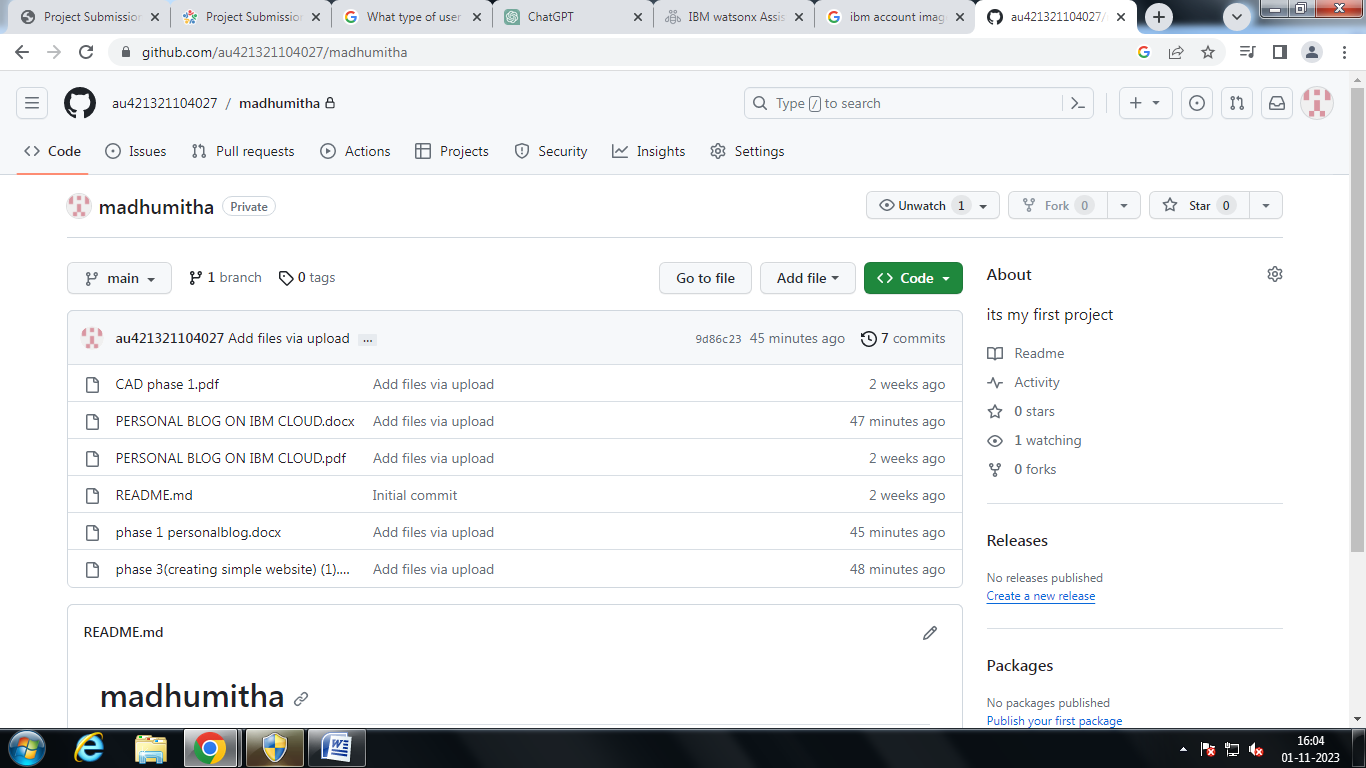
<!-- Add similar links for About and Contact page



**Step1** : **Create a new Static Web App :**

****

**Step 2: Create a New Repository:**

****

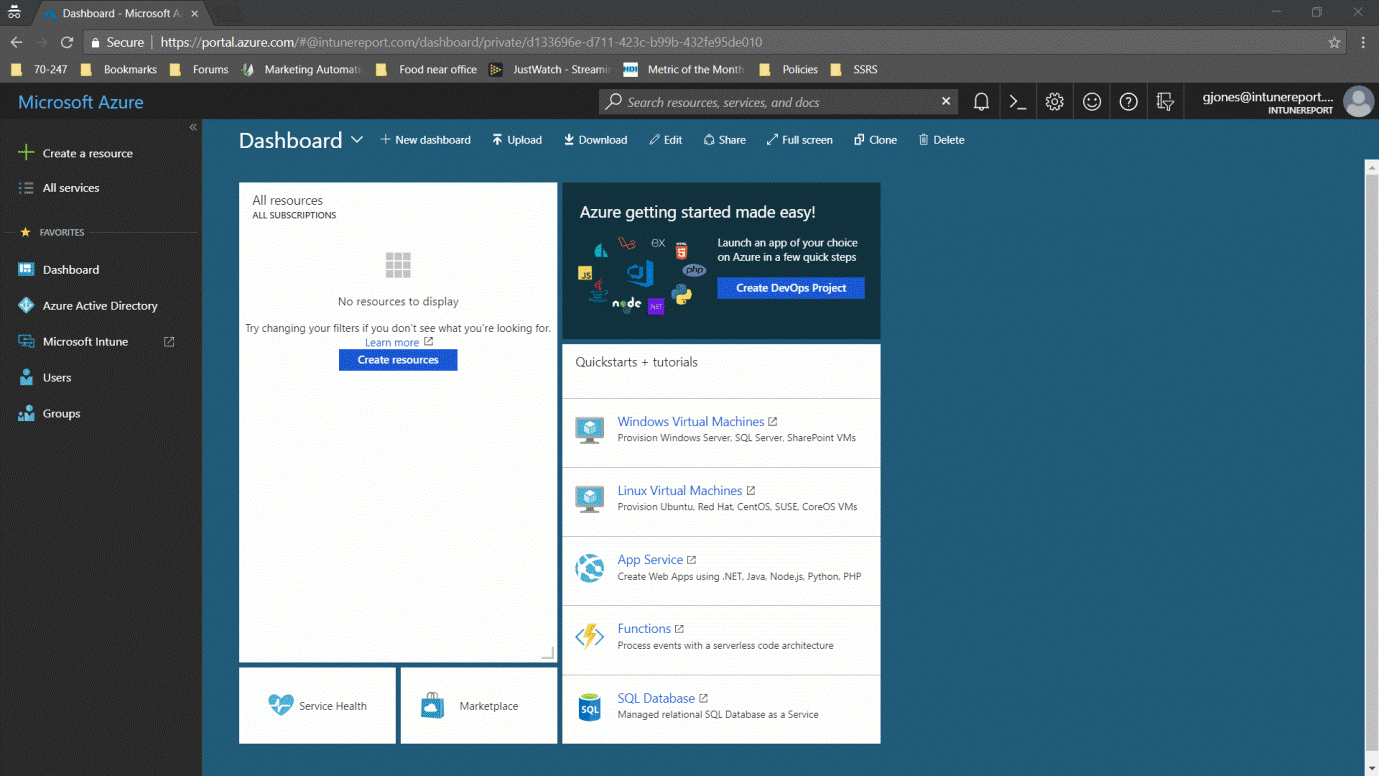
**Step 3: using template create a HTML website :**

****

**Step 4: Azure Portal:**

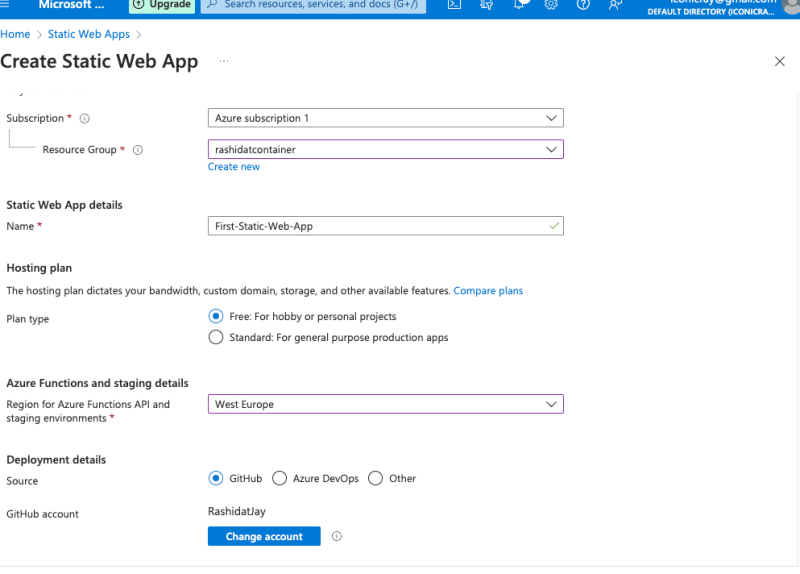
**Log in to the Azure Portal (**[**https://portal.azure.com**](https://portal.azure.com)**).**

1. Go to https://www.azure.com and click the green “Start free” button. ...
2. Next, click another “Start free” button. ...
3. If you already have an account with Microsoft, for example, Microsoft 365, you'll be prompted to log in. ..
4. When you log in, some of your details may already be there.



**Step 5:** **Create a New Static Web App:**

* **Sign in to the Azure portal.**
* **Select Resource groups.**

****