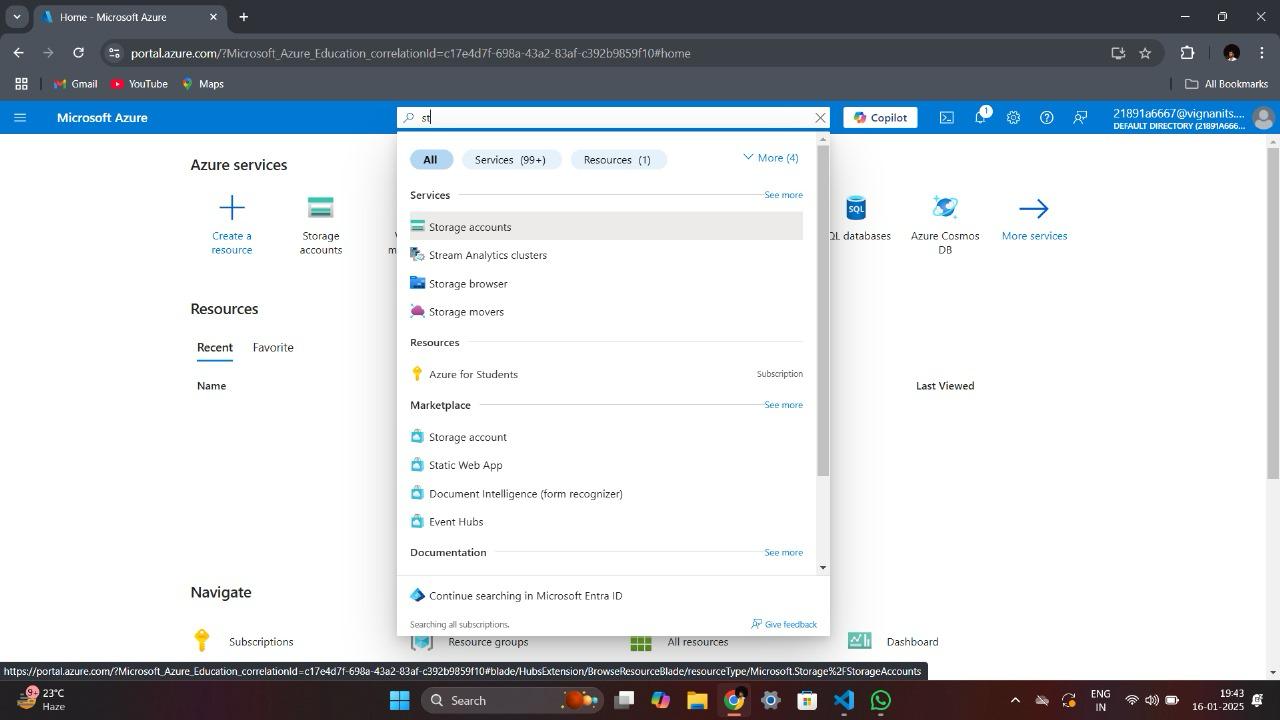
**Documentation: Hosting a Static Website on Azure Storage Account**

Hosting a static website on Azure Storage is a straightforward, cost-effective solution for lightweight web applications like personal blogs, portfolios, or documentation. This guide walks you through the process of creating and deploying a static website using Azure's **Storage Account** service.

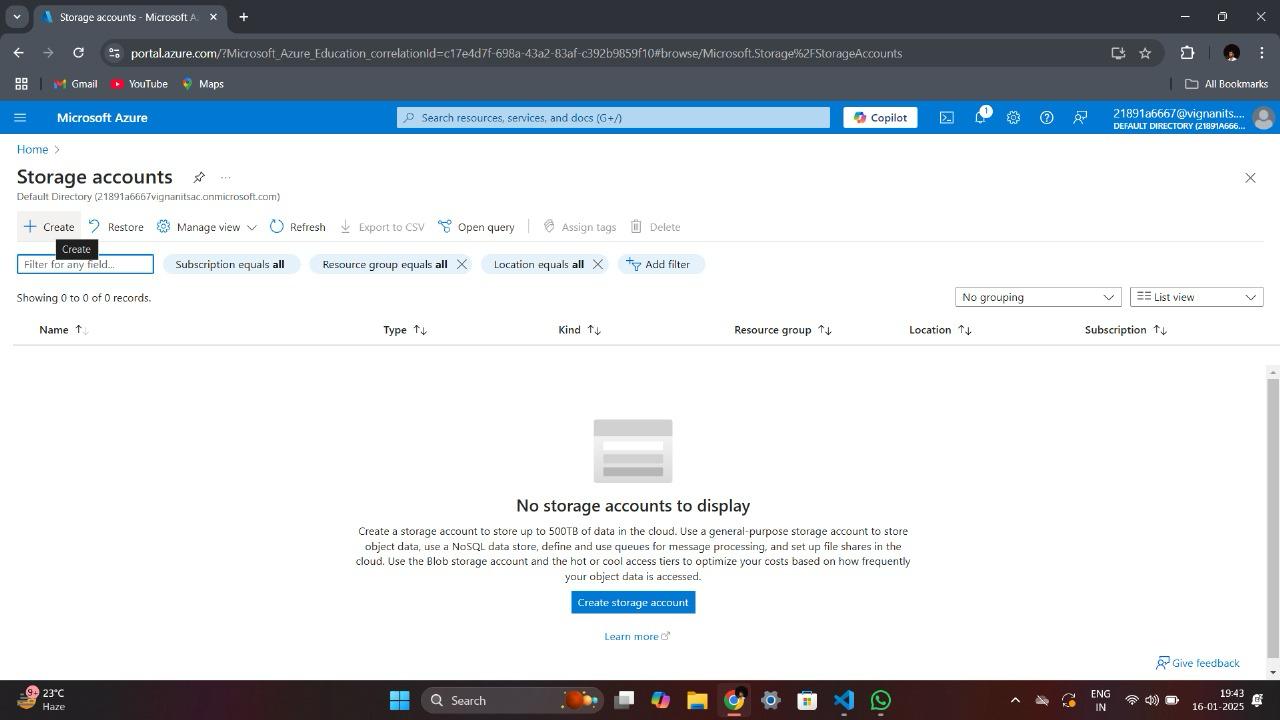
**1. Create a Storage Account**

A Storage Account serves as the foundation for hosting static websites.

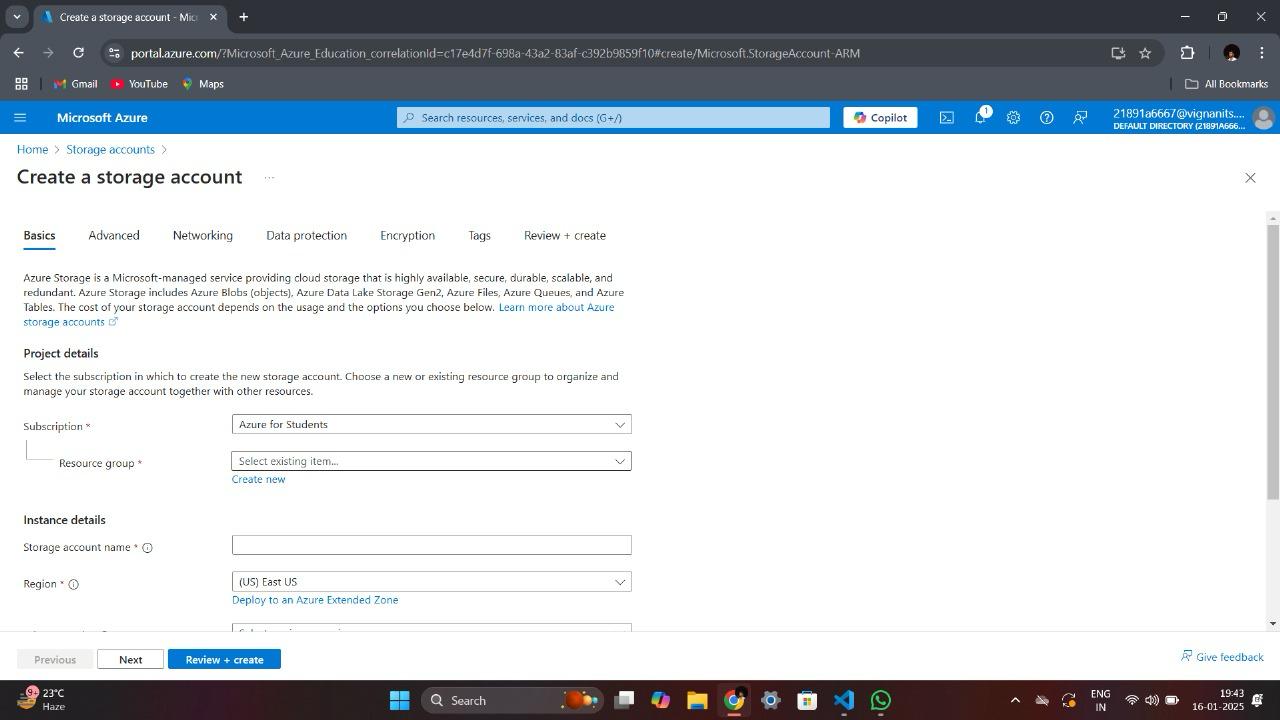
* **Steps to create a Storage Account:**
  1. Open the **Azure Portal** and navigate to **Storage Accounts**.



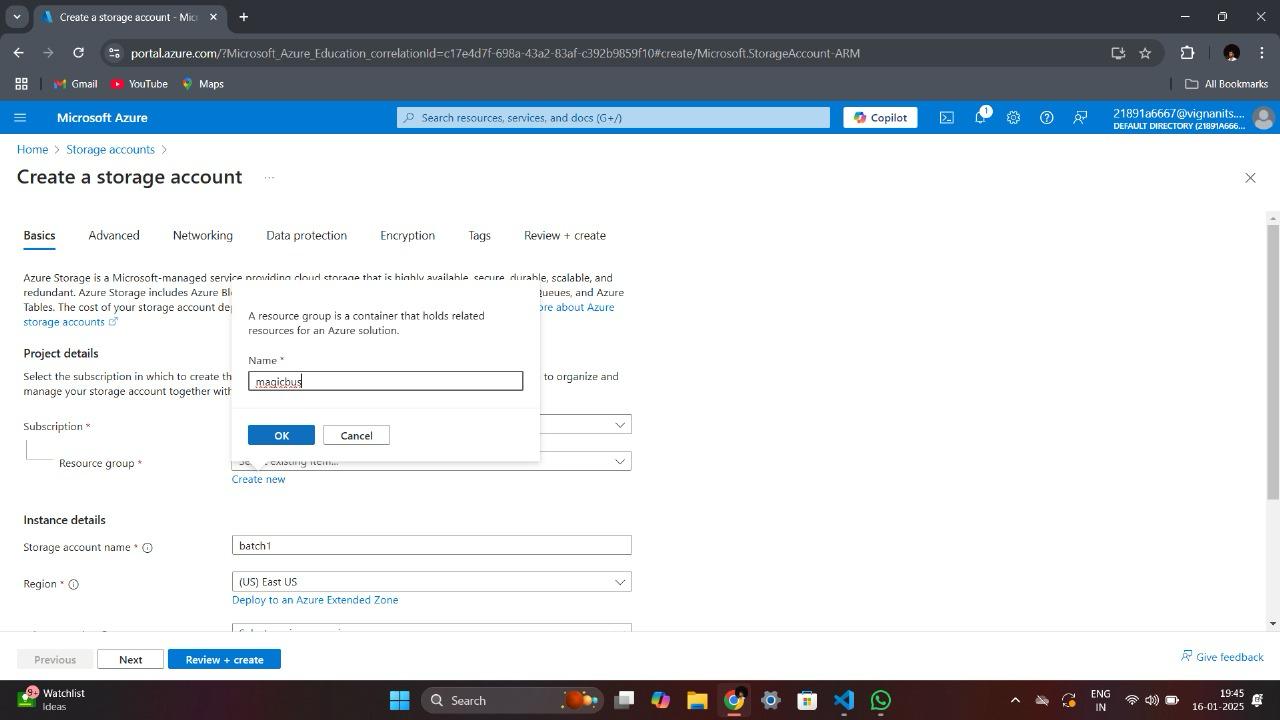
* 1. Click **Create** to start the creation process.



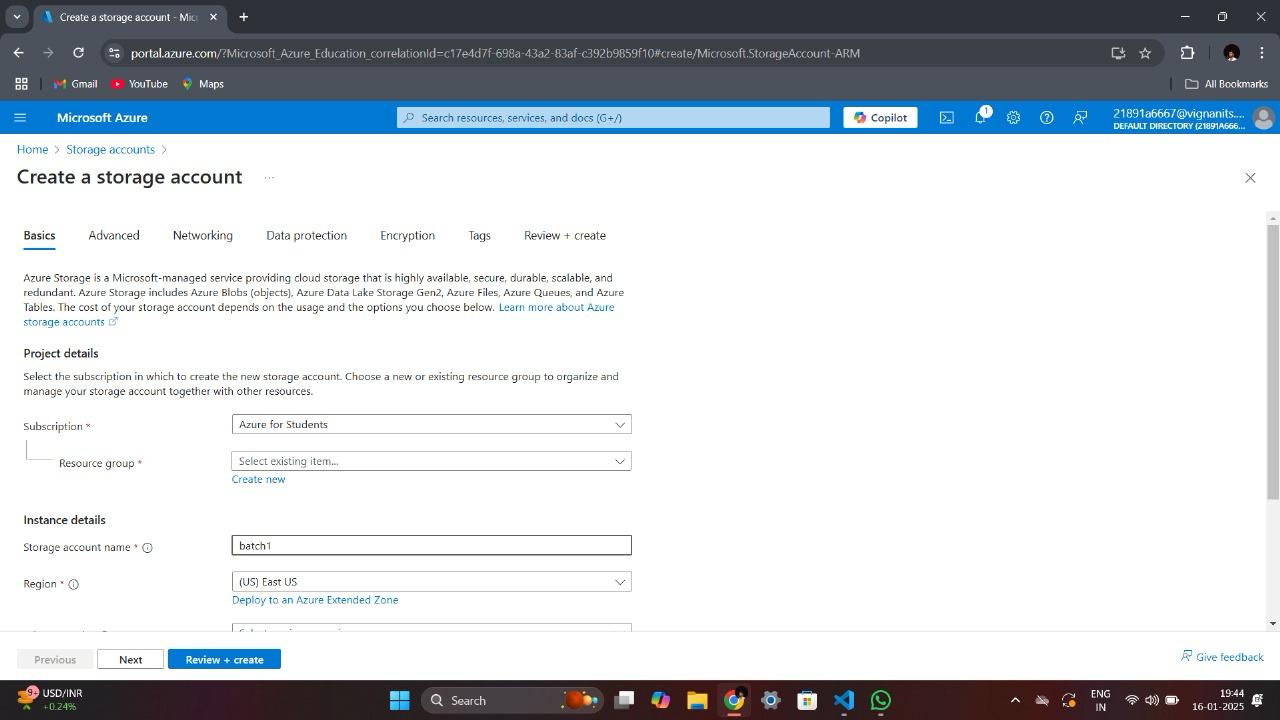
* 1. Fill in the required details:
     + **Subscription**: Select the subscription under which the resource will be billed.



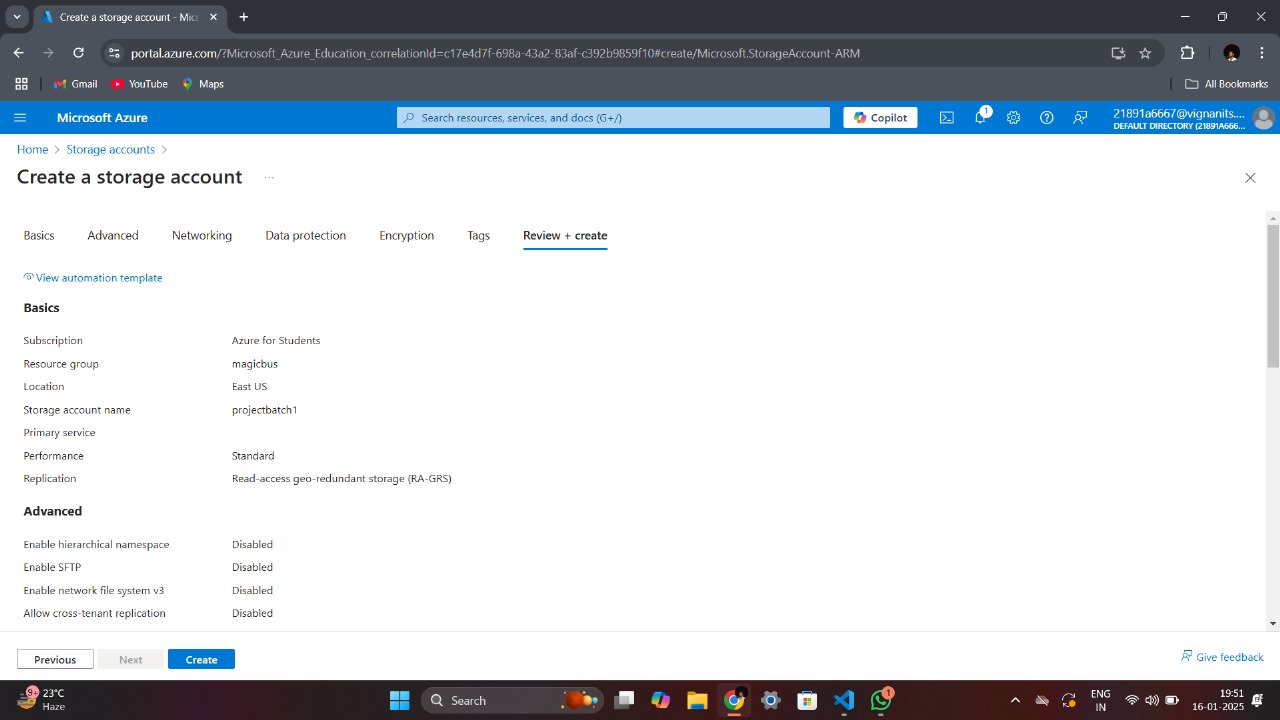
* + - **Resource Group**: Choose an existing resource group or create a new one for better organization.



* + - **Storage Account Name**: Enter a unique name that meets Azure's naming requirements.



* + - **Region**: Select the nearest region to your target audience for reduced latency.
    - **Performance**: Opt for **Standard** performance, which is sufficient for hosting static websites.
    - **Redundancy**: Choose the appropriate redundancy option (e.g., **LRS** for local redundancy or **GRS** for geo-redundancy, ensuring data availability in case of regional failures).
  1. Review your selections, then click **Create** and wait for the deployment to complete.



Once the Storage Account is created, it can be used to store and serve your static website.

**2. Create a Web Page**

To host a static website, you need at least one HTML file (commonly named index.html).

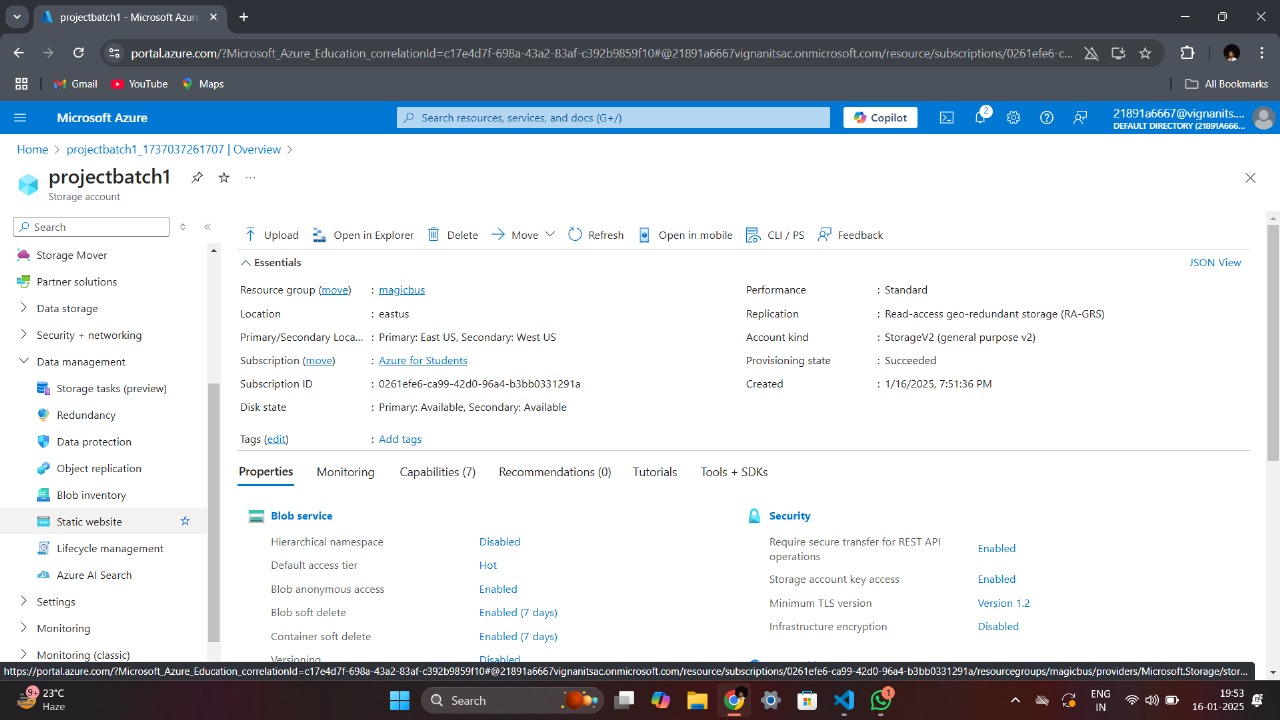
* **Steps to create a basic web page:**
  1. Use any text editor (e.g., Notepad, Visual Studio Code, or Sublime Text) to create an HTML file.
  2. Write simple HTML code with custom text or styling. For example:
  3. <!DOCTYPE html>
  4. <html>
  5. <head>
  6. <link rel="stylesheet" href="project1.css" />
  7. <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css" integrity="sha384-JcKb8q3iqJ61gNV9KGb8thSsNjpSL0n8PARn9HuZOnIxN0hoP+VmmDGMN5t9UJ0Z" crossorigin="anonymous">
  8. <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" integrity="sha384-DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj" crossorigin="anonymous"></script>
  9. <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js" integrity="sha384-9/reFTGAW83EW2RDu2S0VKaIzap3H66lZH81PoYlFhbGU+6BZp6G7niu735Sk7lN" crossorigin="anonymous"></script>
  10. <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js" integrity="sha384-B4gt1jrGC7Jh4AgTPSdUtOBvfO8shuf57BaghqFfPlYxofvL8/KUEfYiJOMMV+rV" crossorigin="anonymous"></script>
  11. </head>
  12. <body>
  13. <div id="sectionHome" class="bg">
  14. <div>
  15. <img height="200px" width="250px" src="https://d1tgh8fmlzexmh.cloudfront.net/ccbp-static-website/software-developer-img.png" />
  16. <h1 class="h">My Projects</h1>
  17. <p class=p>These are few of my static website projects that I have developing using HTML, CSS and Bootstrap</p>
  18. </div>
  19. <div>
  20. <div class=" img d-flex flex-row justify-content-center">
  21. <img onclick="display('sectionTech')" height="80px" width="100px" src="https://d1tgh8fmlzexmh.cloudfront.net/ccbp-static-website/advanced-technologies-img.png" />
  22. <img onclick="display('sectionDiwali')" height="80px" width="100px" src="https://d1tgh8fmlzexmh.cloudfront.net/ccbp-static-website/diwali-img.png" />
  23. </div>
  24. <div class=" img d-flex flex-row justify-content-center">
  25. <img onclick="display('sectionMeal')" height="80px" width="100px" src="https://d1tgh8fmlzexmh.cloudfront.net/ccbp-static-website/food-img.png" />
  26. <img onclick="display('sectionNews')" height="80px" width="100px" src="https://d1tgh8fmlzexmh.cloudfront.net/ccbp-static-website/news-paper-img.png" />
  27. </div>
  28. </div>
  29. </div>
  30. <div id="sectionTech">
  31. <div class="background1 d-flex flex-column justify-content-end">
  32. <div class="container1">
  33. <h1 class="heading1">Advanced Technologies</h1>
  34. <P class="paragraph1">Machinery and equipment developed from the application of scientific knowledge.</P>
  35. <button class="button1 bunty">Learn more</button>
  36. <button onclick="display('sectionHome')" class="button1 bhanu">back</button>
  37. </div>
  38. </div>
  39. </div>
  40. <div id="sectionDiwali">
  41. <div class="diwali-top-section">
  42. <h1 class="diwali-top-section-heading">
  43. Celebrate Diwali with your friends
  44. </h1>
  45. </div>
  46. <div class="diwali-bottom-section">
  47. <div class="d-flex flex-row justify-content-center">
  48. <div class="diwali-card-item">
  49. <img src="https://d1tgh8fmlzexmh.cloudfront.net/ccbp-static-website/lamp-img.png" class="diwali-card-image" />
  50. <h1 class="diwali-card-name">Diwali Air Balloon</h1>
  51. <p class="diwali-card-price">Rs 369</p>
  52. </div>
  53. <div class="diwali-card-item">
  54. <img src="https://d1tgh8fmlzexmh.cloudfront.net/ccbp-static-website/diya-img.png" class="diwali-card-image" />
  55. <h1 class="diwali-card-name">Diwali - Lamp</h1>
  56. <p class="diwali-card-price">Rs 50</p>
  57. </div>
  58. </div>
  59. <div class="d-flex flex-row justify-content-center">
  60. <div class="diwali-card-item">
  61. <img src="https://d1tgh8fmlzexmh.cloudfront.net/ccbp-static-website/firework-img.png" class="diwali-card-image" />
  62. <h1 class="diwali-card-name">Sparklers</h1>
  63. <p class="diwali-card-price">Rs 150</p>
  64. </div>
  65. <div class="diwali-card-item">
  66. <img src="https://d1tgh8fmlzexmh.cloudfront.net/ccbp-static-website/firecracker-img.png" class="diwali-card-image" />
  67. <h1 class="diwali-card-name">Fire Cracker</h1>
  68. <p class="diwali-card-price">Rs 560</p>
  69. </div>
  70. </div>
  71. <button onclick="display('sectionHome')" class="btn btn-primary">back</button>
  72. </div>
  73. </div>
  74. <div id="sectionMeal">
  75. <div class="container2 d-flex flex-column justify-content-end">
  76. <div class="bhanu2">
  77. <h1>Happy Meals</h1>
  78. <p>Discover the best foods over the 1,000 restaurants</p>
  79. <button class="button2">Book Now</button>
  80. <button onclick="display('sectionHome')" class="button2">Back</button>
  81. </div>
  82. </div>
  83. </div>
  84. <div id="sectionNews">
  85. <div class="background3 d-flex flex-column justify-content-end">
  86. <div class="container3">
  87. <p class="paragraph3">NEWS OF THE DAY</p>
  88. <h1 class="heading3">iB Cricket announces world's first Virtual Reality Cricket League</h1>
  89. <P class="paragraph3">iB Cricket and Viu together have announced the World's first virtual reality cricket league featuring 12 international cricketers,who will be competing against each other for the iB Cricket Super Over League title in Mumbai. Iconic cricketers like Virender Sehwag, Suresh Raina, Brendon McCullum will be competing in the league</P>
  90. <button class="button3">Read more</button>
  91. <button onclick="display('sectionHome')" class="button3">Back</button>
  92. </div>
  93. </div>
  94. </div>
  95. <script type="text/javascript" src="https://new-assets.ccbp.in/frontend/content/static-ccbp-ui-kit/static-ccbp-ui-kit.js"></script>
  96. </body>
  97. </html>
  98. Save the file as index.html on your local machine.

Feel free to add additional files like CSS, images, or JavaScript if your site requires them.

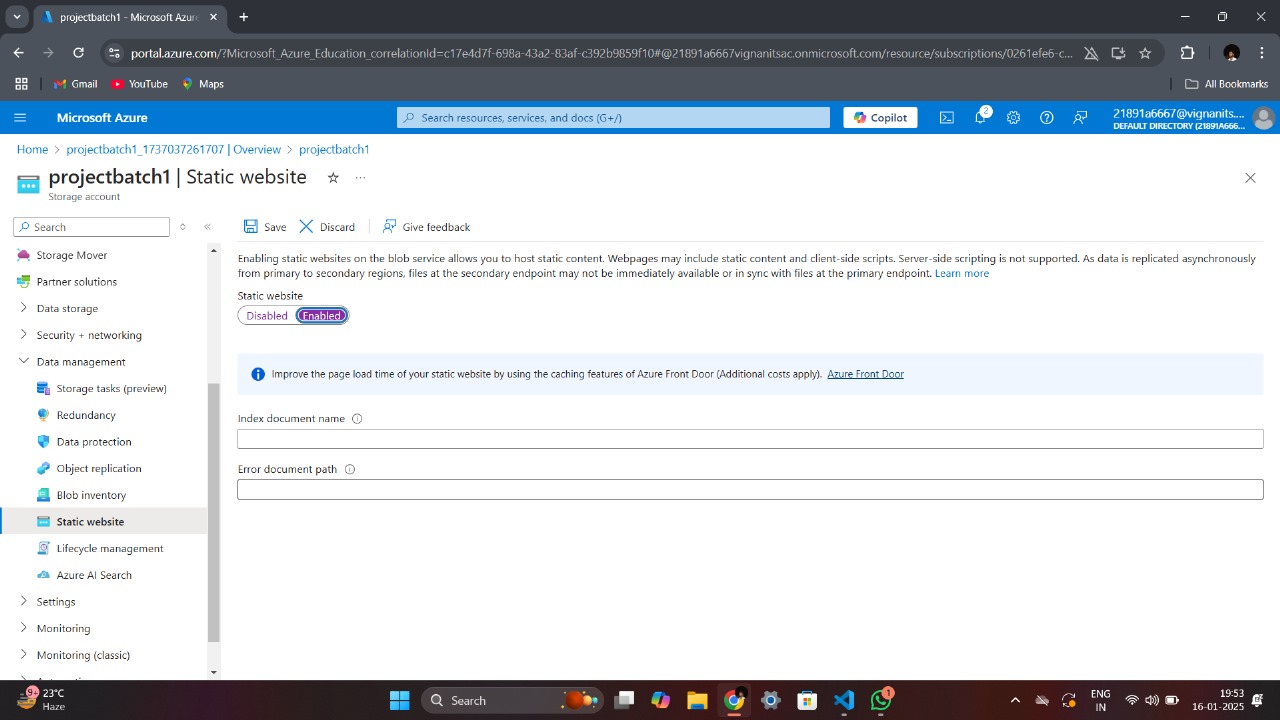
**3. Enable the Static Website Feature**

Azure Storage Accounts offer a dedicated **Static Website** feature to serve web content directly.

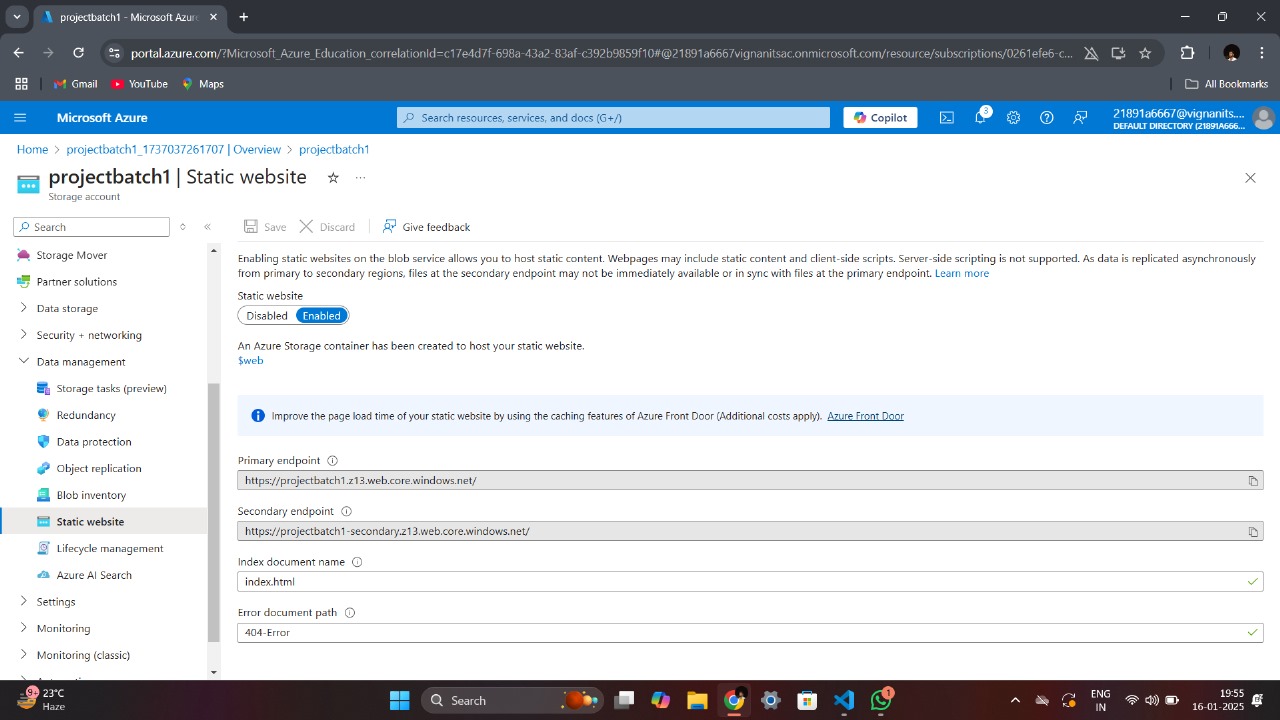
* **Steps to enable the Static Website feature:**
  1. Open the created Storage Account in the Azure Portal.
  2. Scroll down to the **Data Management** section in the left-hand panel and select **Static Website**.



* 1. Click **Enable** to turn on the static website hosting.



* 1. In the **Index document name** field, type index.html (or the main file name of your website).
     + This file will be displayed when users visit the root of your website.



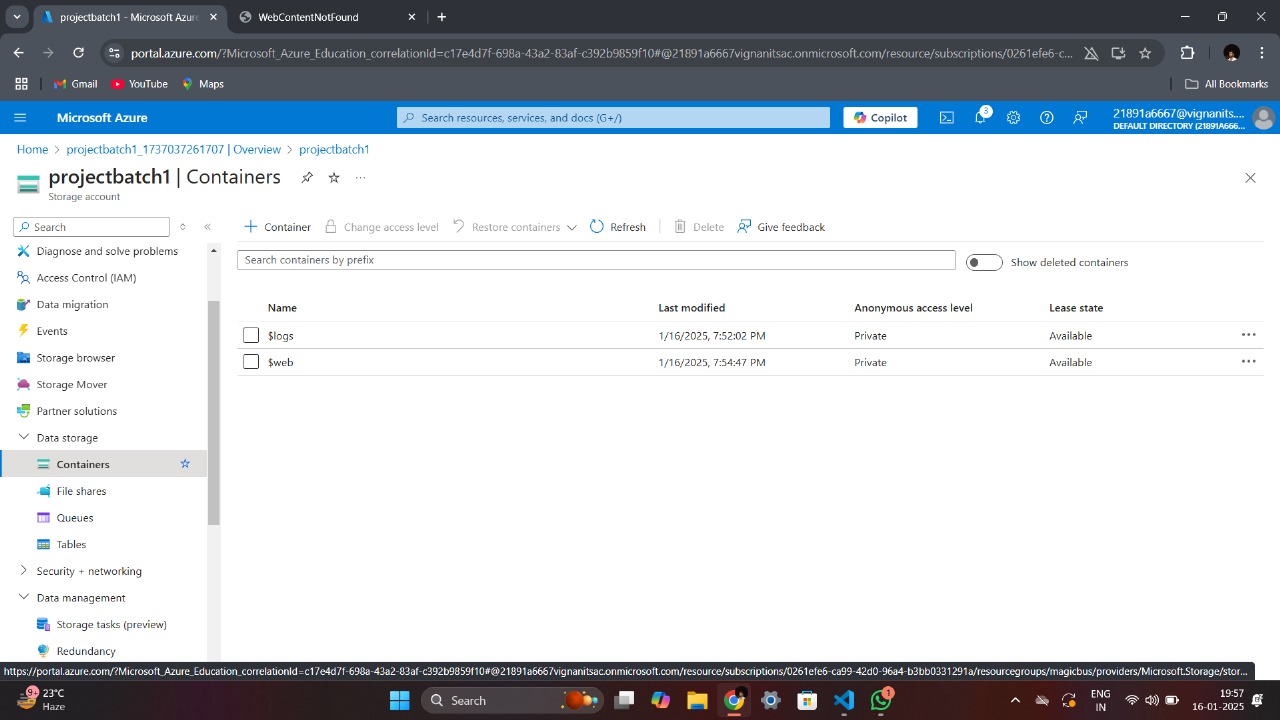
* 1. Save the settings.

Enabling this feature creates a special container called $web, which is reserved for storing website files.

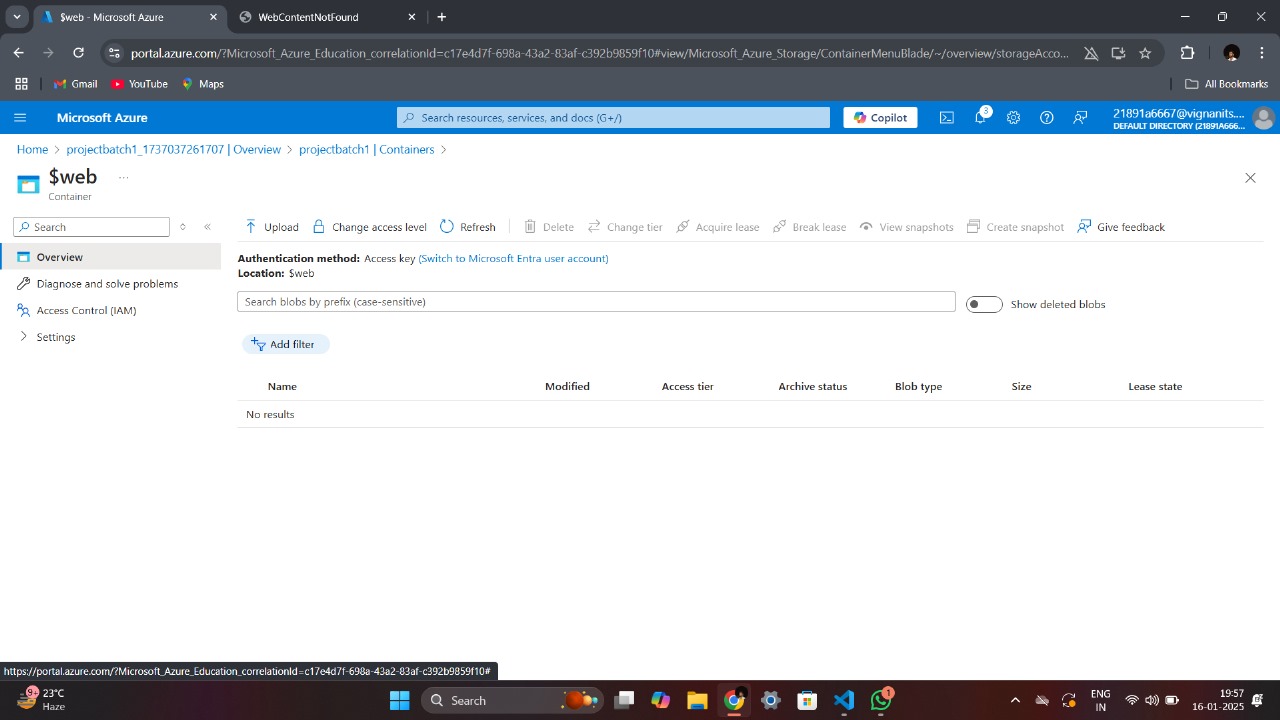
**4. Access the $web Container**

The $web container acts as the directory where your website files are stored.

* **Steps to locate and open the $web container:**
  1. In the Storage Account, go to the **Containers** section under the **Data Storage** menu.



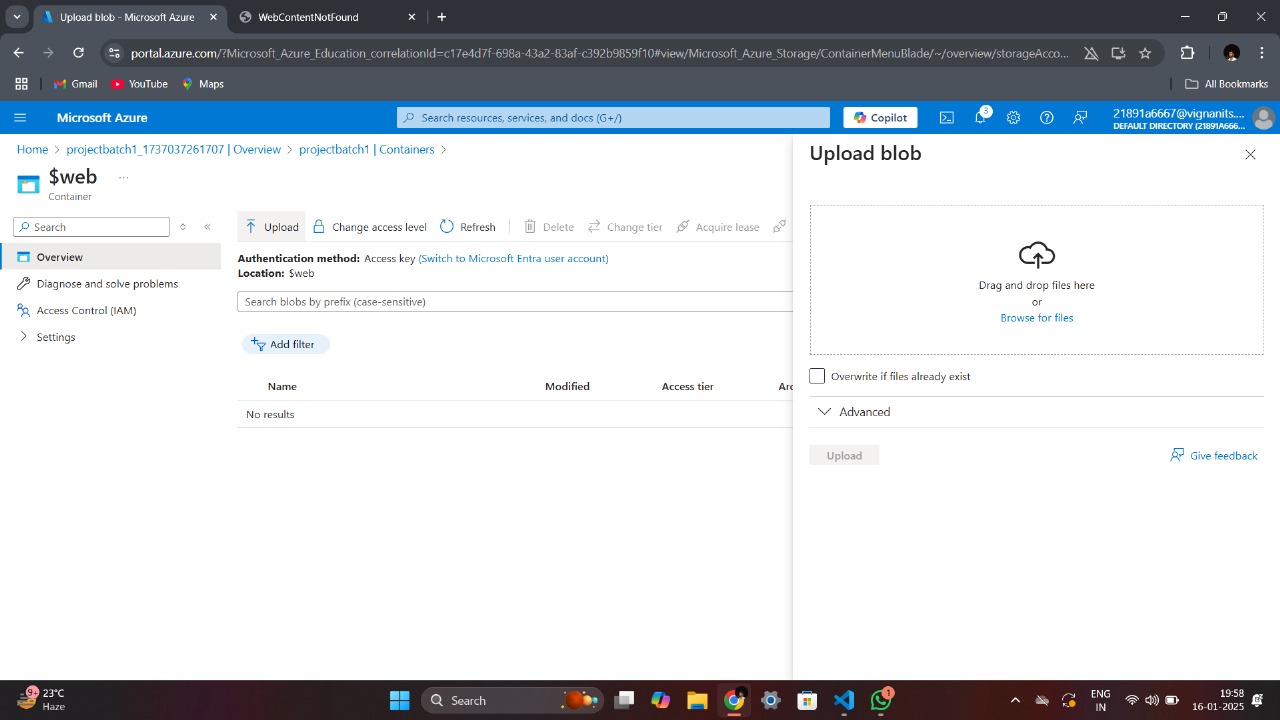
* 1. You will see a container named $web. Click on it to open.



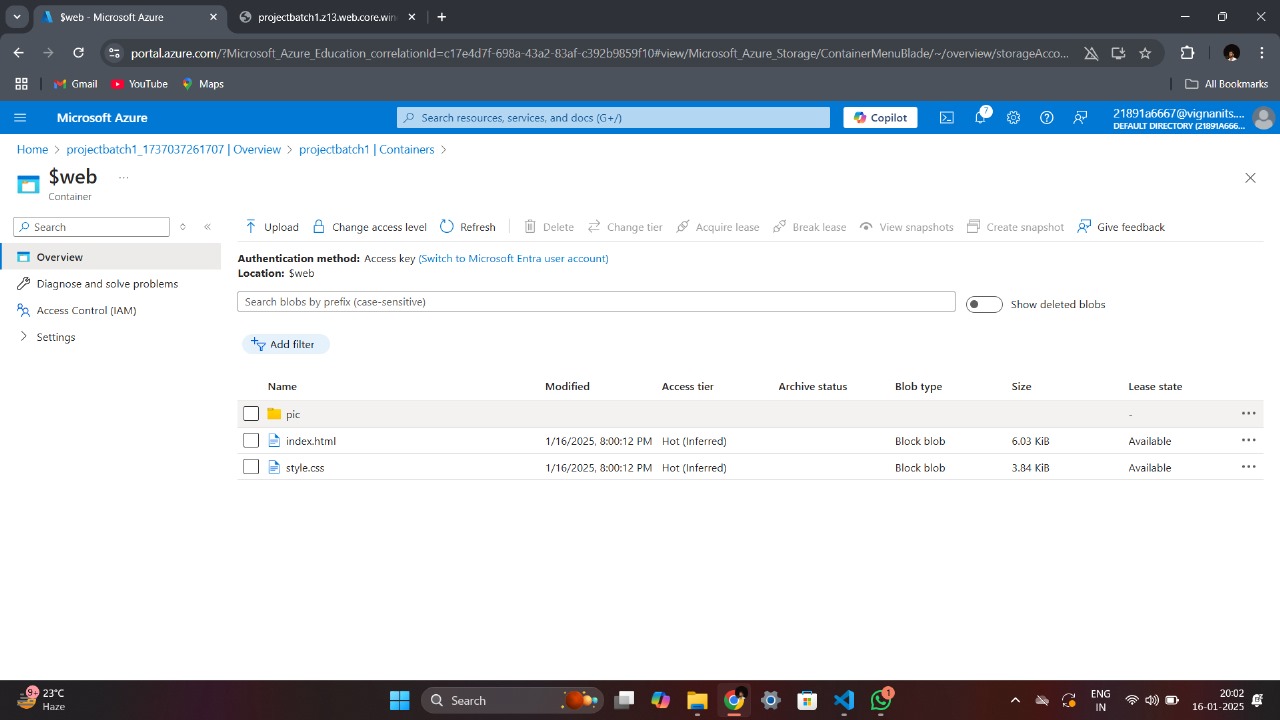
**5. Upload Website Files**

Once the $web container is open, upload your website files.

* **Steps to upload files:**
  1. Click on the **Upload** button within the $web container.



* 1. Use one of the following methods to upload files:
     + **Drag and Drop**: Drag your index.html (and other files, if any) into the upload area.
     + **Browse**: Click the **Browse** button to locate and select files manually.
  2. Click **Upload** to confirm and complete the process.

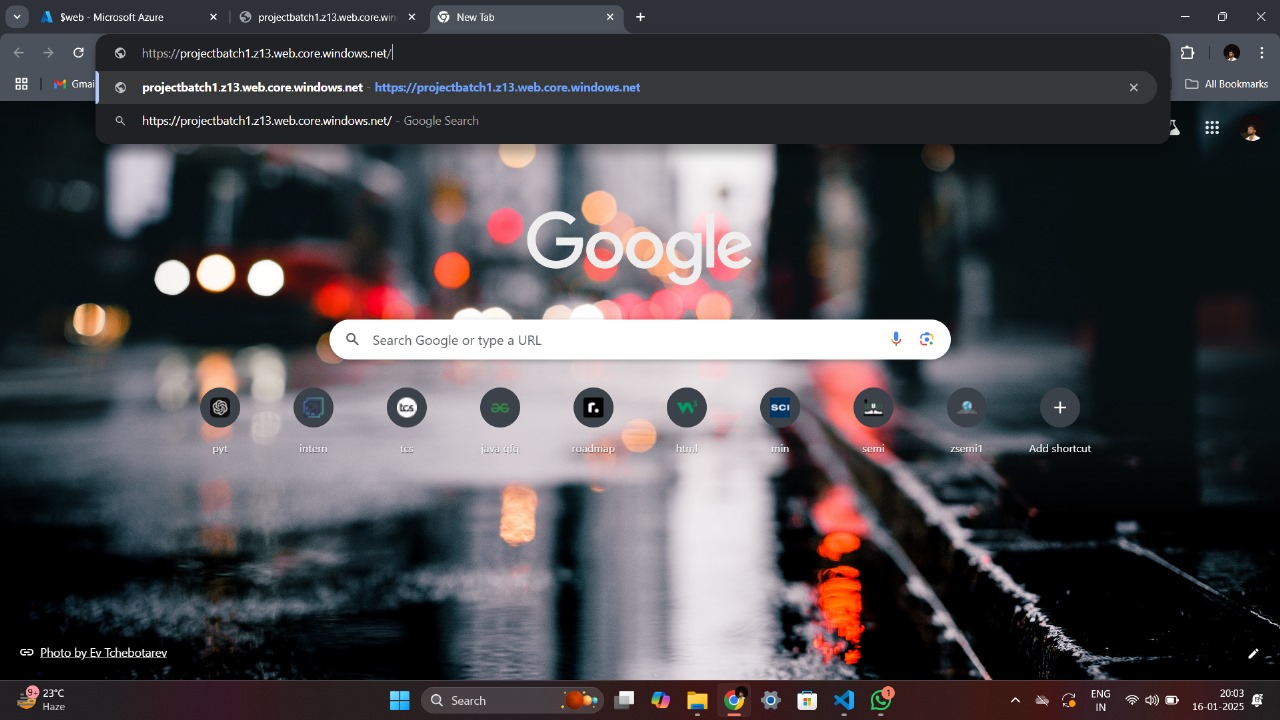


Your website files are now hosted in the $web container and ready to be served.

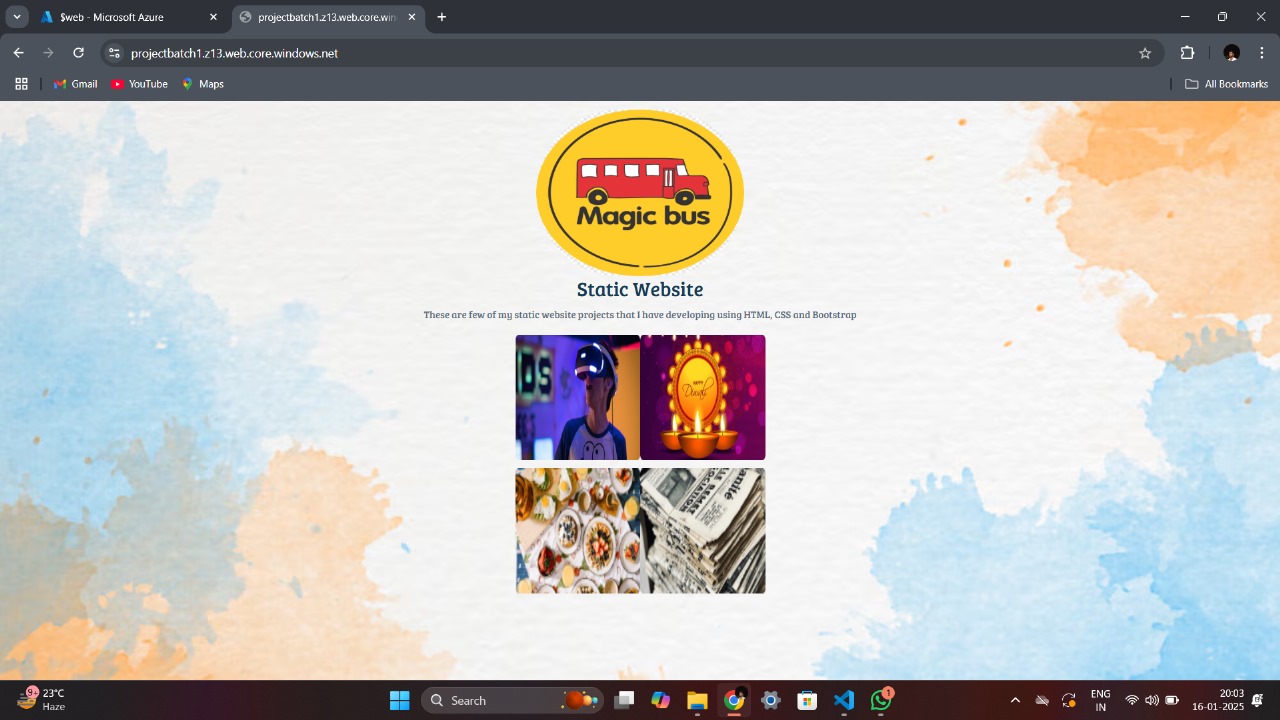
**6. Test the Static Website**

After uploading the files, Azure generates a **Primary Endpoint** URL to access the hosted site.

* **Steps to test your website:**
  1. Go back to the **Static Website** section of your Storage Account.
  2. Copy the **Primary Endpoint** URL displayed there.
  3. Paste the URL into a browser to access your website.



* + - If the website displays correctly, your deployment was successful.
    - If you encounter errors, verify that the file names match the settings (e.g., index.html) and that all dependencies are uploaded.



**Key Azure Services Used**

1. **Azure Storage Account**: The core service for storing and serving static website files.
2. **Static Website Feature**: Enables a container ($web) to host web files and generates a public URL for access.
3. **$web Container**: A dedicated container where all static website files are uploaded.
4. **Azure Portal**: The user-friendly interface for managing and configuring Azure resources.

**Benefits of Using Azure Storage for Static Websites**

1. **Cost-Effectiveness**: Hosting static websites on Azure Storage is significantly cheaper compared to using traditional web hosting solutions.
2. **High Availability**: Azure ensures data redundancy and availability, making your website resilient to failures.
3. **Scalability**: The service can handle traffic spikes without additional configuration.
4. **Simplicity**: No need to manage web servers or complex configurations.
5. **Global Reach**: Azure's globally distributed data centers ensure low-latency access for users worldwide.

**Conclusion**

By following the steps above, you can successfully host a static website on Azure Storage Account. This solution is ideal for lightweight, static content with minimal setup and maintenance.