**Meet Specifications**

**Site:** [**https://tannv-article-cms-project.azurewebsites.net**](https://tannv-article-cms-project.azurewebsites.net)

**Resource Group**

|  |  |
| --- | --- |
| **CRITERIA** | **MEETS SPECIFICATIONS** |
| All relevant resources are contained in a single resource group. | The resource group must include a Storage Account, SQL Server, SQL Database, as well as any relevant services for deploying the web app.  Provide a screenshot of the resource group in Azure, containing your running resources. |
|  | |

**Storage**

|  |  |
| --- | --- |
| **CRITERIA** | **MEETS SPECIFICATIONS** |
| Create and add article data to a SQL Server in Azure. | A SQL Server is created in Azure and is capable of storing the necessary article data (title, author, body).  Provide a screenshot from your SQL database within Azure, showing that both the POST and USER tables have been created. Alternatively, if the site is still live, provide the URL for the site. |
|  | |

|  |  |
| --- | --- |
| **CRITERIA** | **MEETS SPECIFICATIONS** |
| Create and upload images to a Storage Account. | A Storage Account is created in Azure and is capable of storing the necessary image data for the article.  Provide a screenshot from your Storage Account within Azure, with the blob storage endpoint URL visible (can be seen in “Settings”->”Properties”). Alternatively, if the site is still live, provide the URL for the CMS site to show images are able to be stored and viewed. |
|  | |

**Resource Justification**

|  |  |
| --- | --- |
| **CRITERIA** | **MEETS SPECIFICATIONS** |
| Analyze, choose, and justify the appropriate resource option for deploying the app. | In the provided writeup.md  file, for **both** a VM or App Service solution for the CMS app:   * Analyze costs, scalability, availability, and workflow * Choose the appropriate solution (VM or App Service) for deploying the app * Justify your choice   This does not need to be substantially long, but should include information on all four analysis points for each option, your choice, and at least 2-3 sentences on why you choose that option. |
| I choosing a web app service over a virtual machine. Cost-wise, the web app wins hands down with its pay-as-you-go model, saving cash. Plus, you ditch the VM maintenance headache, freeing up your time and energy. Deployment becomes a breeze thanks to the seamless GitHub integration, getting your app live in a flash. High availability is built-in, so your users enjoy consistent uptime. And when it comes to scaling, the web app adapts to your needs effortlessly, avoiding the manual VM configuration struggles. Furthermore, the web app service excels in simplicity and speed. The seamless integration with your GitHub repository allows for effortless deployment with just a few clicks. This streamlines your workflow significantly compared to the cumbersome setup and configuration of a VM. Based on the costs, scalability, availability, and workflow analysis I selected the web app service to deploy for my project article. | |

|  |  |
| --- | --- |
| **CRITERIA** | **MEETS SPECIFICATIONS** |
| Assess app changes that would change your decision. | In the provided writeup.md file, detail how the app and any other needs would have to change for you to change your decision in the last section.  This should be at least 2-3 sentences, but feel free to add as much detail as you feel necessary. |
| I would choose virtual machines when security needs skyrocket and your application architecture explodes with microservices, diverse technologies and user growth, virtual machines (VMs) become attractive allies . Their inherent flexibility protects applications in isolated bubbles, enables granular resource allocation for security, and seamless integration of disparate technologies. Microservices thrive in individual VM houses, easily scaling to meet spikes in user demand. However, keep in mind the cost, management complexity when use virtual machine. | |

**Deployment**

|  |  |
| --- | --- |
| **CRITERIA** | **MEETS SPECIFICATIONS** |
| The Python web app is deployed to Azure. | The Python web app has been deployed to Azure using the chosen resource in the previous section.  As evidence, provide a screenshot of the Python application running from a browser (this can be part of the screenshot in the next section). **The screenshot should include the URL and the black header that states “Article CMS”.** Alternatively, you can provide a link to the deployed app, if it is still live. |
|  | |

|  |  |
| --- | --- |
| **CRITERIA** | **MEETS SPECIFICATIONS** |
| The Python web app is able to connect to storage. | The Python web app is able to connect to the related storage solutions.  As evidence, provide a screenshot of the Python application running from a browser. **The screenshot should include the URL and at least one article containing title, author, body, and an image.** Alternatively, you can provide a link to the deployed app, if it is still live. |
|  | |

**Security & Monitoring**

|  |  |
| --- | --- |
| **CRITERIA** | **MEETS SPECIFICATIONS** |
| Add a functioning “Sign in with Microsoft” option to the app. | The Python web app has an additional, operational option to sign in with Microsoft.  As evidence, provide a screenshot of the redirect URIs configured within the App Registration page in Azure. Alternatively, you can provide a link to the deployed app, if it is still live.  Additionally, your code in views.py should appropriately implement the Microsoft sign-in button using the msal library. |
|  | |

|  |  |
| --- | --- |
| **CRITERIA** | **MEETS SPECIFICATIONS** |
| Access attempts to the app are logged. | Both successful and unsuccessful attempts to access the web app are logged.  As evidence, provide a screenshot or download the logs from Azure containing at least one successful and one unsuccessful access attempt, and include in your submission files. If otherwise submitting a URL, please include a link to screenshot/logs in the “Submission Details” box on the project submission page. |
|  | |