**3. CLOUD COMPUTING COMPONENTS USED**

**3.1 Application**

The core application of our project is a recommendation system that provides company suggestions for amateur investors based on customer risk profiles. Built using Python, Flask, HTML, and CSS, the application processes customer data input via a web form, computes investment risk scores using pandas, and generates a list of companies that match the user's investment profile. This application is hosted on the cloud to ensure scalability, accessibility, and ease of deployment.

**3.2 Platform**

Our platform is based on cloud services, enabling easy management of infrastructure and application deployment. We use Flask as the web framework for building and running the recommendation engine, allowing integration with cloud services. Additionally, Terraform is utilized as our Infrastructure as Code (IaC) tool to automate the setup, provisioning, and management of cloud resources, ensuring a streamlined deployment process.

**3.3 Storage**

The project stores customer data and company information in Excel files, which are processed by the backend to generate recommendations. Cloud storage is utilized for securely storing and retrieving these files, ensuring data persistence and easy access for the application. By leveraging cloud-based storage, we ensure data availability and scalability without the need for local infrastructure.

**3.4 Infrastructure**

The infrastructure is provisioned and managed using Terraform. This includes the creation of necessary virtual machines, networking, and other resources required to run the web application. By utilizing cloud infrastructure, we benefit from the flexibility, scalability, and cost-efficiency offered by cloud service providers, allowing us to adjust resources based on application demand.

**3.5 Deployment**

Deployment of the application is fully automated using Terraform. By defining infrastructure as code, we ensure that the entire deployment process is repeatable and consistent across environments. The application is deployed on cloud servers, which host the Python Flask app and handle user interactions via the web interface. The deployment leverages the flexibility of the cloud to ensure high availability and minimal downtime.

These sections provide a clear overview of the cloud components used in your project, focusing on how each element contributes to the overall functionality.

 **Cloud Computing Overview**  
Amazon Web Services. (n.d.). What is Cloud Computing?  
Available at: <https://aws.amazon.com/what-is-cloud-computing/>

 **Infrastructure as Code (IaC) with Terraform**  
HashiCorp. (n.d.). Introduction to Terraform.  
Available at: https://learn.hashicorp.com/terraform

 **Flask Framework Documentation**  
Pallet Projects. (n.d.). Flask Documentation.  
Available at: https://flask.palletsprojects.com/en/latest/

 **Cloud Storage Explained**  
Google Cloud. (n.d.). Cloud Storage Overview.  
Available at: https://cloud.google.com/storage/docs/overview

 **Cloud Deployment with Terraform**  
DigitalOcean. (2022). How To Manage Infrastructure with Terraform.  
Available at: https://www.digitalocean.com/community/tutorial\_series/how-to-manage-infrastructure-with-terraform

 **Automating Cloud Infrastructure**  
Microsoft Azure. (n.d.). Deploy Infrastructure Using Terraform.  
Available at: <https://learn.microsoft.com/en-us/azure/developer/terraform/>