**Azure-Powered To-Do List Web App**

The app will be a simple to-do list with user authentication, task management, and task categorization.

1. **Requirements:**

**Functionalities:**

* **User Authentication:**
  + Users can sign up, log in, and manage their profiles.
* **Task Management:**
  + Users can create, update, delete, and mark tasks as completed.
* **Categories:**
  + Users can organize tasks into categories (e.g., Work, Personal, Shopping).
* **Task Prioritization:**
  + Users can assign priority levels to tasks (e.g., High, Medium, Low).
* **Basic Dashboard:**
  + Users can view a summary of tasks by category and priority.

**Storage Needs:**

* Storage Needs: The relational database will manage structured data, including user accounts, task details, and task categories. It will also ensure that the schema supports these entities and relationships.
* File Storage: This service will manage unstructured data such as documents or images.

**Security Requirements:**

* Data Encryption: The application should ensure that the data is encrypted while stored in Azure SQL database and Azure Blob Storage using the Azure’s built-in encryption mechanisms.
* Access Control: The application should manage User Authentication and Authorization securely and it should control the access to Azure Resources.
* Secret Management: The application should be able to securely store and manage sensitive application settings, connection strings and cryptographic keys and it should be able to retrieve secrets securely at runtime.
* Network Security: The application should be able to isolate application resources and control traffic flow and it should protect against unauthorized access and secure network traffic.

**Scalability Considerations:**

* Auto-scaling: The application should be able to scale the resources depending on the varying traffic loads automatically.
* Load Balancing: The application should be able to distribute the traffic across multiple instances to increase performance and reliability.

1. **Design Architecture:**

Frontend:

* Azure App Service: Host a simple HTML/CSS/JavaScript frontend. This platform will provide a managed environment for deploying and scaling the frontend of your web application.

Backend:

* Azure App Service: Host backend APIs using Node.js or another simple backend framework. This platform will handle the server-side logic and API requests.

Database:

* Azure SQL Database: Use for relational data storage, including user accounts, tasks, and categories. It provides a managed database service with built-in high availability and scaling.

Storage:

* Azure Blob Storage: Utilize for storing static assets (like images and documents) or optional user-uploaded content. Blob Storage is scalable and cost-effective for handling large amounts of unstructured data.

Networking & Security:

* Azure Virtual Network (VNet): Isolate resources within a secure network environment.
* Network Security Groups (NSGs) and Azure Firewall: Secure the network by controlling inbound and outbound traffic and protecting against potential threats.

Identity Management:

* Azure AD B2C: Manage user authentication and authorization securely, allowing users to sign up, log in, and manage their profiles.

Secrets Management:

* Azure Key Vault: Securely store and manage sensitive data, such as connection strings and API keys. Integrate with your application to access secrets securely at runtime.

Domain Management:

* Azure DNS: Configure for a custom domain name, providing users with a user-friendly URL for accessing your web application.

Monitoring & Insights:

* Azure Monitor: Set up to gather insights into the performance and health of your application. This tool will help you track metrics, set alerts, and diagnose issues across your Azure resources.