SIDDHI S. KOTRE
ROLL NO: 50

Distributed System and Cloud Computing Lab
Azure Practical

Aim: Develop application for Microsoft Azure.

Theory: Microsoft Azure is a comprehensive cloud computing platform that provides a wide range of cloud services, including computing power, storage, databases, networking, analytics, and more. Developing applications for Azure allows businesses and developers to leverage these powerful cloud resources to create scalable, flexible, and cost-effective applications.

When developing applications for Azure, you can utilize various services and tools, such as **Azure App Services**, **Azure Functions**, **Azure Kubernetes Service (AKS)**, **Azure Virtual Machines (VMs)**, and much more. Below is an overview of key concepts, tools, and steps involved in developing applications for Microsoft Azure.

Cloud Computing Basics

- Cloud Models: Azure provides different service models, including IaaS (Infrastructure as a Service), PaaS (Platform as a Service), and SaaS (Software as a Service). Applications can be developed using any of these models depending on the requirements.
  - IaaS: Offers infrastructure resources like VMs and networking.
  - PaaS: Provides application development and hosting services, like Azure App Services and Azure Functions.
  - SaaS: End-user applications hosted on the cloud, such as Microsoft Office 365.

Azure Services Overview

- Compute Services: For running your application logic and hosting your services.
  - Azure App Services: PaaS offering for hosting web apps and APIs.
  - Azure Functions: Serverless computing service to run event-driven code.
  - Virtual Machines (VMs): IaaS for running full virtualized instances of operating systems.
- Storage Services: For storing data securely and reliably in the cloud.
  - Azure Blob Storage: Object storage for unstructured data like images and videos.

  - Azure SQL Database: Managed relational database service based on Microsoft SQL Server.
  - Azure Cosmos DB: Globally distributed NoSQL database.
- Networking Services: To manage network traffic, load balancing, and virtual networks.
  - Azure Virtual Network (VNet): Networking service for connecting virtual machines and services securely.
  - Azure Load Balancer: Ensures high availability by distributing incoming network traffic across multiple resources.
- Security Services: Protect your application and data.
  - Azure Active Directory (Azure AD): Identity and access management service.
  - Azure Key Vault: Safeguards cryptographic keys, secrets, and certificates.
- DevOps and Monitoring Services: For continuous integration and deployment, as well as performance monitoring.
  - Azure DevOps Services: Provides version control, build, and release management.
  - Azure Monitor: Monitors application and infrastructure health and performance.

Steps to Develop an Azure Application in Visual Studio 2022

1. Install Visual Studio 2022

- Download the Visual Studio 2022 installer from the official Visual Studio website.
- Run the installer and ensure the following workload is selected:
  - Azure Development
- Click Install or Modify (if Visual Studio is already installed).
- Wait for the installation to complete, then launch Visual Studio.

2. Create a New Project

1. Open Visual Studio 2022.
2. Click Create a new project on the start screen.
3. Configure the following options:
   - Language: C#
   - Platform: Azure
   - Project type: Cloud
4. Choose Azure Cloud Service (extended version).
   - If the option is unavailable, click More tools → Azure development → Modify to ensure the required components are installed.

3. Configure the Project

1. Click Next.
2. Set the Project name (e.g., AzureCloudService1).
3. Choose the desired location for the project.
4. Click Create.

4. Set Up the Azure Cloud Service

1. In the New Microsoft Azure Cloud Service (Classic) window:
   - From the left-hand side, select Visual C# → ASP.NET Web Role.
   - Click the > button to add the role to your service.
2. Click OK.

5. Create an Empty Project Template

1. In the next window:
   - Choose the Empty project template for simplicity.
2. Click Create.

6. Add a Web Form

1. Open Solution Explorer (if it’s not visible, go to View → Solution Explorer).
2. Locate WebRole1 in the project structure.
3. Right-click WebRole1 → select Add → Web Form.
4. Name the Web Form (e.g., Default.aspx) and click OK.

7. Run and Debug the Project
   1. Click Debug → Start Debugging (or press F5).
   2. Wait for the application to build and launch in the browser.
   3. Observe the output of the project running on the local Azure emulator.

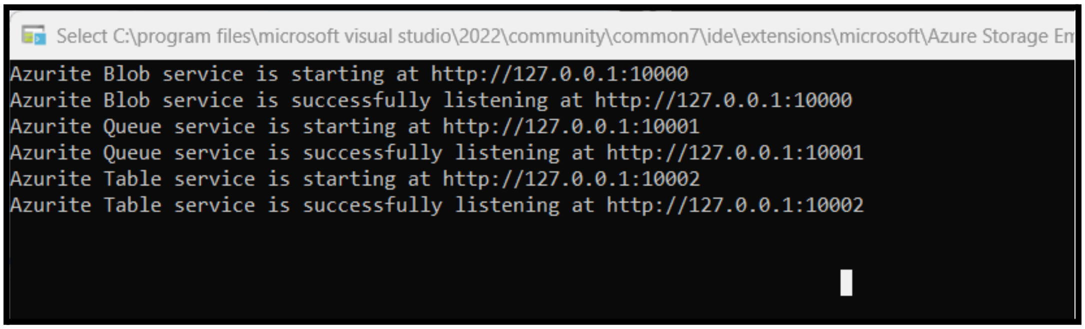
Code:

WebForm1.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs"
Inherits="WebRole1.WebForm1" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
  <title>Cloud Application Development</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      background-color: #f4f4f4;
      margin: 0;
      padding: 20px;
    }
    h1 {
      color: #333;
    }
    .container {
      background-color: white;
      padding: 20px;
      border-radius: 5px;
      box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);
    }
    button {
      margin-top: 10px;
      padding: 10px 15px;
      background-color: #007BFF;
      color: white;
      border: none;
      border-radius: 5px;
      cursor: pointer;
    }
    button:hover {
      background-color: #0056b3;
    }
  </style>
```

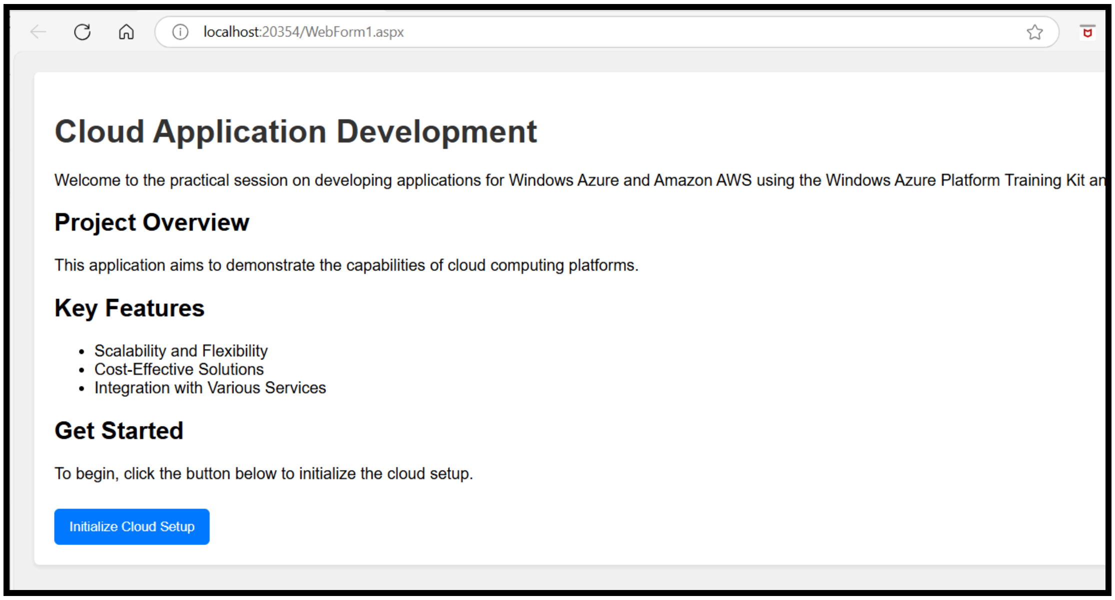
```
</head>
<body>
  <form id="form1" runat="server">
    <div class="container">
      <h1>Cloud Application Development</h1>
        <p>Welcome to the practical session on developing applications for Windows Azure and
Amazon AWS using the Windows Azure Platform Training Kit and Visual Studio.</p>

      <h2>Project Overview</h2>
            <p>This application aims to demonstrate the capabilities of cloud computing
platforms.</p>

      <h2>Key Features</h2>
      <ul>
        <li>Scalability and Flexibility</li>
        <li>Cost-Effective Solutions</li>
        <li>Integration with Various Services</li>
      </ul>

      <h2>Get Started</h2>
      <p>To begin, click the button below to initialize the cloud setup.</p>
           <button type="button" onclick="alert('Cloud setup initialized!')">Initialize Cloud
Setup</button>
    </div>
  </form>
</body>
</html>
```

Output:



Select C:\program files\microsoft visual studio\2022\community\common7\ide\extensions\microsoft\Azure Storage Em

```
Azurite Blob service is starting at http://127.0.0.1:10000
Azurite Blob service is successfully listening at http://127.0.0.1:10000
Azurite Queue service is starting at http://127.0.0.1:10001
Azurite Queue service is successfully listening at http://127.0.0.1:10001
Azurite Table service is starting at http://127.0.0.1:10002
Azurite Table service is successfully listening at http://127.0.0.1:10002
```



Cloud Application Development

Welcome to the practical session on developing applications for Windows Azure and Amazon AWS using the Windows Azure Platform Training Kit an

Project Overview

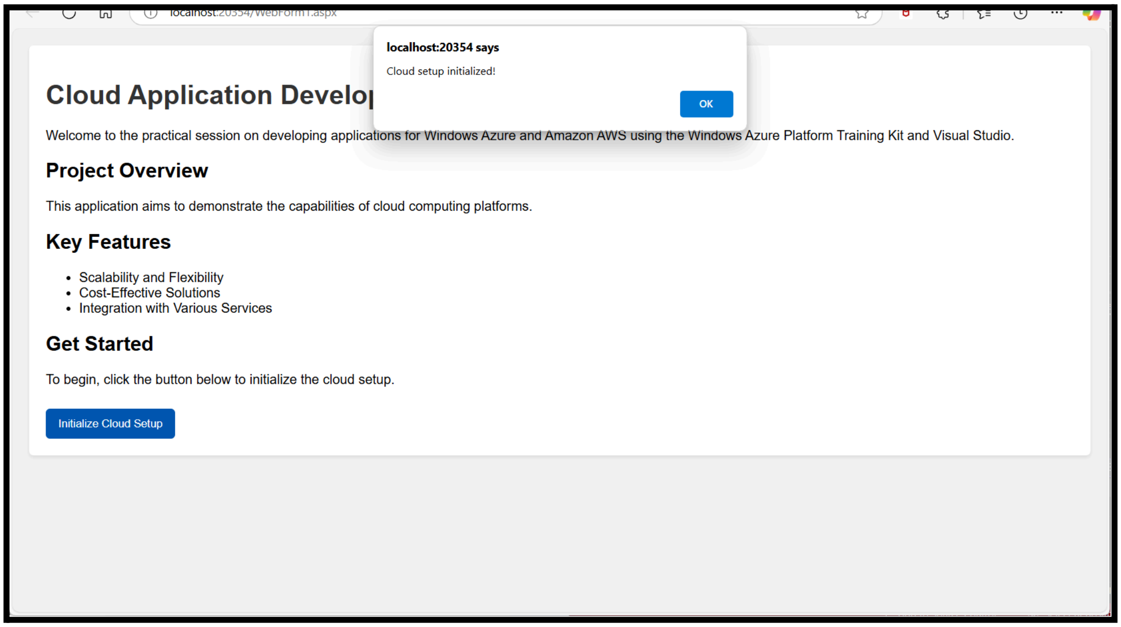
This application aims to demonstrate the capabilities of cloud computing platforms.

Key Features

- Scalability and Flexibility
- Cost-Effective Solutions
- Integration with Various Services

Get Started

To begin, click the button below to initialize the cloud setup.



localhost:20354 says

Cloud setup initialized!

Cloud Application Develop

Welcome to the practical session on developing applications for Windows Azure and Amazon AWS using the Windows Azure Platform Training Kit and Visual Studio.

Project Overview

This application aims to demonstrate the capabilities of cloud computing platforms.

Key Features

- Scalability and Flexibility
- Cost-Effective Solutions
- Integration with Various Services

Get Started

To begin, click the button below to initialize the cloud setup.