🏠 Homework 2 – Host a Simple Web App in Azure App Service

🎯 Objective

Build and deploy a simple **dynamic web application** in **Azure App Service**.
Your application must expose **a homepage** with:

- an **input text field**
- an **“Enter” button**
- a **persistent list of items** displayed below

Each submitted item must be stored in a **managed Azure database**, and **the database must not be publicly accessible** — only your App Service is allowed to connect.

You may choose any tech stack supported by Azure App Service (Node.js, Python, .NET, Java, PHP…).
You may use AI tools (ChatGPT, Copilot, etc.) to help generate code.

💡 Tips

- Keep the app extremely simple — a single page is enough.
- Use App Service’s **Configuration → Application Settings** to store connection strings securely.
- For database security:
    - Azure SQL → Firewall rules must allow *only* your App Service’s outbound IPs
    - Cosmos DB → Public access disabled; allow *only* your App Service’s outbound IPs
- Use `/api` endpoints, templating engines, or any basic framework you know — simplicity is fine.
- Deployment scripts don’t need to be sophisticated; they just need to work.

🧩 Assignment Requirements

1. Application Requirements

Your application must:

1. Have a publicly accessible homepage (no authentication required).
2. Display:
    a. A text input field
    b. An “Enter” button
    c. A list of previously submitted items
3. On each submission:
    a. Store the item in a **managed Azure database**:
        i. Azure SQL Database
        or
        ii. Azure Cosmos DB (Core API)
4. The list must remain consistent even after app restarts / browser refresh → data must be **persisted** in the database.
5. UI can be minimal — focus is on functionality and correct use of Azure services.

2. Security Requirement (Mandatory)

Your database **must not be publicly accessible**.
Your App Service must be the **only allowed client**.

Depending on DB choice:

If you choose Azure SQL Database

- **Disable “Allow Azure Services”**
- **Do NOT allow your personal IP** (except temporarily while debugging)
- Add **only the App Service outbound IP addresses** to the SQL Server firewall
  → find them in **App Service → Properties → Outbound IP Addresses**

If you choose Azure Cosmos DB (NoSQL)

- **Disable public access** (Public network access = Disabled)

- Create a **Network Rule** allowing:
    - **App Service outbound IP addresses**
- No other IPs should be able to access the account

You will be graded on whether your database is truly **locked down**.

3. Deployment Requirements

You must deploy the app to **Azure App Service** using an automated script (`deploy.sh` or `deploy.ps1`).

Your **deployment script** must:

1. Create the **App Service Plan**
2. Create the **App Service**
3. Create the **database** (SQL or Cosmos DB)
4. Apply **database network restrictions**
5. Configure application settings (connection strings, keys, etc.)
6. Deploy your application code
    a. Via ZIP deploy
    b. Or via Az CLI `az webapp deploy`

⭐ Optional Bonus — Deploy Using Infrastructure as Code (IaC)

For extra credit, you may provision all required Azure resources using **Infrastructure as Code (IaC)** technologies such as **Terraform**, **ARM Templates**, or **Bicep**.

To earn the bonus, your IaC configuration must create every resource your application needs (App Service Plan, App Service, database, networking rules, configuration settings, etc.), and your deployment script (deploy.ps1 or deploy.sh) must run your IaC code before deploying your application code. The entire deployment — from infrastructure creation to application publishing — must be reproducible simply by running your script.

4. Deliverables (ZIP File)

You must submit a **ZIP** containing:

a. Source Code

Your complete app source code:

- Backend application
- HTML/CSS/JS files (if applicable)
- Package descriptors (`requirements.txt`, `package.json`, `.csproj`, etc.)

b. Deployment Script

A single script:

- `deploy.sh` **or**
- `Deploy.ps1`

`(+ IaaC in case you go for the bonus)`

c. README.txt (or README.md)

Must include:

1. **Your full name**
2. **Public URL of the deployed web app**
3. **Technology stack used** (Node/Python/.NET/etc.)
4. **Database choice** (SQL / Cosmos DB)
5. **Security explanation:**
    a. How you restricted DB access
    b. Which IPs you allowed
6. **Any instructions** needed to test your app

📨 Submission Instructions

Deadline: *Sunday, 14.12.2025, 23:59*

You must:

1. Upload a **private GitHub repository**

- **a.** Add me as contributor: **StefanNedelcu**
- **b.** Place everything in a folder named **TEMA-2**

2. Send me:
   - a. The **ZIP file** in a private Microsoft Teams chat
   - b. A link to your GitHub repo

Both GitHub and the ZIP must contain the same content.