Deployment Document

Aim: Our main aim is to deploy our Plaboo project in the production environment, without any issues. After this general description, the details of how exactly our web application is to be published to Azure have been discussed.

Hosting Domain	Azure instance
Production link	https://plaboo.azurewebsites.net/
Database Details	Azure RDBMS (SQL)
Software used	Visual Studio 2019, SQL Server 2014 Management Studio

Database:

- In Visual Studio 2019 inbuilt RDBMS, create a database schema and export the csv files into the created tables to insert data. Use Code first approach to migrate the tables (Steps given in README on github)
- Create Azure database instance in SQL Server 2014 Management Studio and import the same csv files/use insert statements to fetch the data on production.

Front-end:

- Used HTML to make the basic layout. Used CSS to design and add transition effects to the HTML page.
- Used Javascript and jquery to retrieve dynamic data from the database and show it on the screen.
- Used bootstrap to make our website responsive for mobile phones.

Back-end:

- Used ASP .NET MVC framework to develop the backend.
- Used C# language to develop the back-end.

Deploy an ASP.NET app to Azure with Azure SQL Database

Azure App Service provides a highly scalable, self-patching web hosting service. This document elaborates on how we deployed our data-driven ASP.NET app in App Service and connected it to Azure SQL Database.

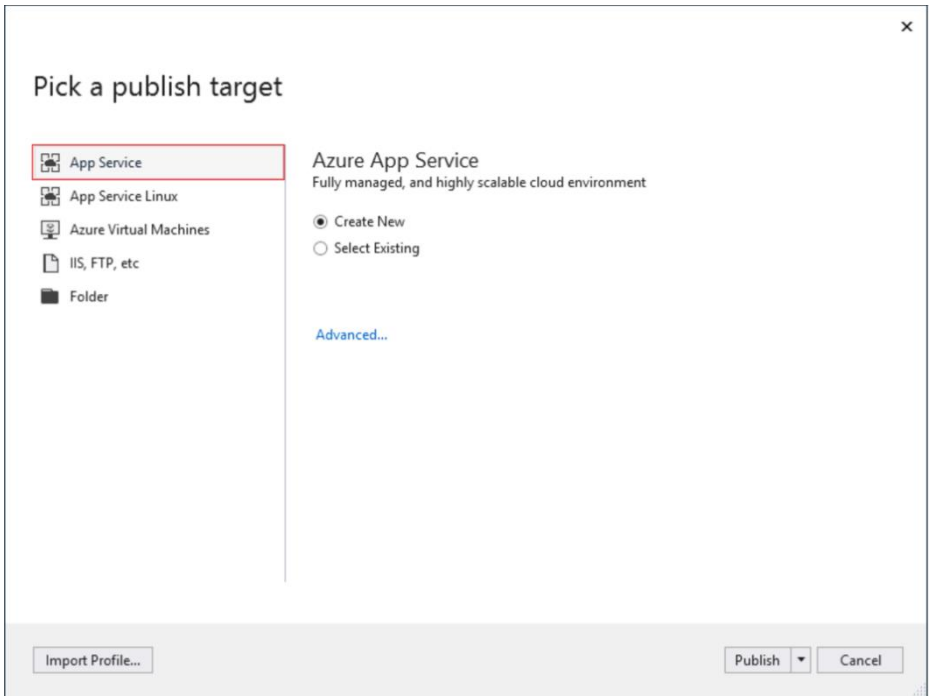
1. Prerequisites

Install Visual Studio 2019 with the ASP.NET and web development workload. Next, create a free student Azure account.

The app uses a database context to connect with the database. In our web application we have used a connection string named DBConnection. The connection string is set in the Web.config file and referenced in the Context/PlabooContext.cs file. The connection string name is used later to connect the Azure app to an Azure SQL Database.

2. Publish ASP.NET application to Azure

In the Solution Explorer, right-click our application and select Publish.



3. Sign in to Azure

In the Create App Service dialog, click Add an account, and then sign in to your Azure subscription. If you're already signed into a Microsoft account, make sure that account holds your Azure subscription.

4. Next, create a resource group and configure the web app name

5. Create an App Service plan

   An App Service plan specifies the location, size, and features of the web server farm that hosts your app. You can save money when hosting multiple apps by configuring the web apps to share a single App Service plan.

   1. Next to App Service Plan, click New.
   2. In the Configure App Service Plan dialog, configure the new App Service plan

6. Next, Create a server

   Before creating a database, you need a logical SQL server. A logical SQL server is a logical construct that contains a group of databases managed as a group.

   1. Click Create a SQL Database and click a new SQL Server as well.

7. Next, Create a database in Azure SQL Database

8. Once all the required resources are created, publish your web application. Once the wizard finishes creating the Azure resources, it publishes your ASP.NET app to Azure. Your default browser is launched with the URL to the deployed app.