PupBuddy – CA2 – Oana Cozma & Sorin Avel

Project Documentation

1. **Azure SQL Database**

The Azure Database was created via the Microsoft Dashboard, this included activating the student subscription for Azure, creating the Server the database would be sitting on and creating the resource group. The connection string was made available for the MVC project & creator I.P. addresses added to the firewall settings.

1. **MVC Application**

We created our C# MVC Application without authentication in Visual Studio by using the default template to which we initially added the models – PuppyModel & MeetingModel, each with constructors for each property and a default primary key. Following this we created the Database Context (PuppyContext) which inherits from DbContext and has DbSet for each of the tables that we would need creating in the database, with constructors for each. The connection string was then added to WebConfig file – this was taken from the Azure database that was created ahead of the project (at the same time as creating the Server it is sitting on).

Next we created the required controllers (for Puppy and Meetings), which contain methods for displaying the list of items in the database for the respective table, as well as having CRUD capabilities via the other methods.

Finally the views were altered and edited as needed to implement the functionality required for the web app, each section of if (Puppy/Meetings) having their own Index & CRUD pages.

1. **WebAPI**

We added the WebAPI to the existing MVC project firstly by installing 2 NuGet packages (Microsoft.Asp.Net.WebApi.Core & Microsoft.Asp.Net.WebApi.WebHost). Afterwards we created a registration file for the API in App\_Start – WebApiConfig.cs which sets the route for the web app. Then Global.asax.cs needed to have System.Web.Http referenced so that was added as well as a GlobalConfiguration setting.

Next we created the API Controller and amended its settings to be able to get the list from the PuppyModel table sitting within the Azure database.

Swagger was also installed in the project to serve API testing & documentation.

1. **Console Client App**

To test the API we created a separate solution & project for the C# Console App where the Puppy class is declared – with its properties and constructors. ToString is then used to print the individual rows of the table. The Program class connects to the API URI which then uses a Try/Catch in a For loop.