The project is an ASP.NET MVC project.

https://dotnet.microsoft.com/en-us/apps/aspnet/mvc

The app is based around a few main models:

ApplicationUser.cs

Hunt.cs

Location.cs

These are migrated to a SQLite database.

Can be browsed with DB Browser for SQLite.

https://sqlitebrowser.org/dl/

Application User:

The application user is authenticated through a default Microsoft authentication service.

The actual object stored in the database is not stored exactly as the ApplicationUser.cs model.

Stored in the AspNetUsers table, it adds password hashes and other extra auth features.

Hunt:

Hunts are created by the admin and accessed by Users through an AccessCode.

The AccessCode is stored as {users-phone-number}/{hunt-name} for a particular user.

The HuntLocation model/table is an intermediate table that links the Hunt and Location table due to the one-to-many relationship between Hunt and Location.

When creating a hunt, the admin can enter html or just normal text for an invitation text. This message will be sent as an email and SMS notification to a user when they are added to a hunt, if html is entered, it will be displayed in the email. The SMS is currently hard coded to Verizon, but the functionality should already be there to select whatever carrier the user chose on registration, we just have not tested it because we all had Verizon. So to test other carriers, remove line 277 in HuntController.cs.

Adding tasks to the hunts is a little convoluted with the many screens you must go through, but when adding a task, it is then added to a list of all tasks. Then you navigate back to the hunt you want to add it to and hit the add task button the click the link next to that task you want to add in the master list of tasks.

Location:

The Location model is what defines the task for a hunt. Currently it has properties for latitude and longitude, but there is no current GPS functionality set up. There is also no QR code functionality setup but there is a property for it on the class. So, the only current use for the location class is to store question and answer tasks which are displayed on a hunt.

The way a task is tracked as completed for an individual user is that the ApplicationUser has a list of completed tasks associated with them. When the hunt is displayed it gets the current user and checks if the tasks are in that list. If they are, they are displayed as completed.

Other miscellaneous information:

QuestionController.cs and Questions View

- Not sure exactly what this is used for, to our knowledge it is not utilized in any place currently. Seems to be an attempt to implement GPS functionality.

CSS

- Stored in wwwroot in the CSS folder.

Login and Register

- Has validation for the inputs.
- The user is automatically logged in if the access code for that user for a specific hunt is entered.