Put it Together

# Problem Scenario

An office on campus needs you to help them build a form for users to register for an event. They would like to capture the name of the person registering, their email address, their physical address, whether they are a CMU faculty, staff, or student or not affiliated with CMU, and if they have any food allergies for the meal to be served as well as any extra comments/questions the person may have. They also host multiple events through the year so they need the user to select which event that he or she may be registering for. The office would like to be able to have reports for events that show the registered attendees (ordered by last name, first name for easy location during registration sign in) along with a separate report for dining services related to the food allergies for those who have indicated they have them. A confirmation email should be sent upon successful registration and the administrators should have the ability to send a reminder email for events if they desire (i.e. bulk sending to all registrants).

# Model the Data

Create the data model required to solve this problem. Do you need more than one table? If there is more than one table what relationships are required? Think about the entities involved and create a simple ER diagram or data model to help you think about and support the requirements. Select appropriate data types and lengths of fields for this data.

# Initial HTML

Begin with the Bootstrap tutorial at W3Schools.com: <http://www.w3schools.com/bootstrap/default.asp>

This introduction will familiarize you with Bootstrap principles and this is important because Bootstrap forms the layout framework for our web sites. After completing this, create a new ASP.NET MVC solution (you will need Windows Authentication even though all pages should not be authenticated). Implement the application template (ask a member of the design team to assist with this so you have the current application template).This will be similar to the MVC 1 assignment/tutorial. Make sure to use Bootstrap for this project (within the context of the CMich application template).

# Databases

Databases (we use SQL Server 2008 and 2012, primarily) allow us to have a consistent way to store and load lots of data across many applications in the same manner, along with all sorts of benefits for backups, transactions, etc.

Most of our applications work with a database of some sort. Many times, you will just use Entity Framework to access the data (similar to the MVC 1 tutorial). Most often, however, you will need to be comfortable creating and managing databases along with the queries and stored procedures used to read and write data.

Create the necessary tables in your own test database. You should use EntityFramework, but should stick with the database-first approach, for the typical data operations against the tables but you should explicitly create stored procedures for the reports (not that you will always, but as a way to demonstrate proficiency). Create appropriate indexes to support reporting needs and relationships/keys to support proper data management practices.

# Site Requirements

Allow the user to register and when the user signs in have a simple list of “My Registrations.” Your site should have a page that is used by the registrants as well as a confirmation page (the email should also be sent, but the user should see a confirmation of registration in the browser as well). Use principles of web form design to lay out the page in an accessible and responsive manner. (<http://channel9.msdn.com/Events/MIX/MIX10/EX03> is a presentation you may watch to brush up on principles). There are some built in components within ASP.NET to assist with managing the users (look at identity related topics).

There should be an administrative page/area as well. This should allow admins to add new events to the list to register for, provide for reporting by event (two reports previously described), have the functionality to remove registrants from an event, and permit the administrator to send a reminder email for the event. This email should include the registrant’s name and event related information (like a template). This administrative portion should be secured using Windows Authentication to the two user groups of CENTRAL\IT-CheckWeb and CENTRAL\IT-StudentAuthors.

For sending of email, use CMich.Settings.SystemSettings.DefaultMailServer (found in the CMich.Settings assembly) for the SMTP host. You must use a valid “From” address so you can use your own or autohelp@cmich.edu for that.

Please create a web.config [transformation](http://www.asp.net/mvc/overview/deployment/visual-studio-web-deployment/web-config-transformations) so that when the site is published in Release build configuration (vs. the default Debug configuration) that custom errors are turned on and point the database connection string to a different server (for the purposes of this assignment just point it to a dummy server, it doesn’t have to be an actual production server).

Finally, in order to facilitate error logging, etc. check out [ELMAH](https://code.google.com/p/elmah/) (also available via NuGet) and here is a [blog article about ELMAH](http://www.hanselman.com/blog/ELMAHErrorLoggingModulesAndHandlersForASPNETAndMVCToo.aspx). Please implement this and have SQL logging to a COMMON database location (i.e. in common with other students on IT-SQL2k8Test server in Elmah database). ELMAH is an area of experimentation so ask questions as needed.