project 2

April 20, 2020 Arlington .NET / Mark Moore

- Angular single-page application
    - client-side validation
    - error handling on requests to APIs
    - deployed to Azure App Service
    - supports deep links
- ASP.NET Core REST service
    - follow standard HTTP uniform interface, except hypermedia
    - architecture with separation of concerns between domain/business logic, data access, and API; repository pattern
    - deployed to Azure App Service
    - Entity Framework Core
        - DB should be on the cloud
        - all DB/network access should be async
    - server-side validation
    - support filtering or pagination on at least one resource
    - logging
    - implement hypermedia, or, implement an API Description Language, e.g. using Swashbuckle
    - (optional: implement a custom filter, health check, or middleware, e.g. exception-handling middleware)
- Azure Pipelines
    - CI pipelines
        - Unit tests
        - SonarCloud
            - Code coverage at least 50% for API, at least 50% for Angular app
            - Reliability/Security/Maintainability scored at least B
    - (optional: deploy in release definition or separate job instead of in build job, and use health checks)

- must have some calls to an external API, or integration with some other service
- authentication and authorization with Auth0 or Okta
- Scrum processes
  - Project board to track user stories across team.
  - Standup at least two or three times a week
- any other tech you want within reason
- the data model (how many tables, what kind of complex relationship like N to N) must be at least as complicated as project 1.
- the user interaction model (what are the user stories, what inputs/interactions can the user make) must be at least as complicated as project 1.
- a project proposal
  - MVP minimum viable product
  - potentially stretch goals, extra features