

SAT Application Requirement Summary

1. Create a Trello board to track the progress of the project. Create cards for individual tasks and assign the cards to the team member who will complete the work. Ensure cards are detailed and reference necessary resources.
2. Assign a team member to take screenshots of the Trello board as the project progresses. This team member should also get screen shots of interesting sections of code. These screen shots will be used for project documentation.
3. Build a relational database for the application based on the schema discussed in PMD.
4. Add data to the database. All data should be work appropriate.
5. The project history is tracked using Git and the code base is stored in a public repository on Github. (Note: ensure that you are not pushing usernames or passwords)
6. Create an ASP.NET Web Application for the UI layer of your project.
7. Implement authentication using Identity Samples.
8. Select a multi-page template, download the files, and bring them into an Archive folder in your UI project.
9. Create the data layer and utilize Entity Framework to create domain models for your database objects.
10. Update the connection strings in the web.config of your UI layer.
11. Copy all of the template resources (JS, CSS, Fonts, etc) out of the archive folder and paste them into the appropriate folder in the UI layer.
12. Rename the _Layout to _OriginalLayout and then utilize the appropriate HTML file from the template to create a new _Layout.
13. Update all navigation links and image file paths.
14. Update CSS and JS bundles to include all of the resources for the template. Remove any bundled resources that will not be used. Link to the Styles and Scripts bundles on the _Layout page.
15. Build metadata buddy classes in the data layer.
16. Scaffold out all controllers and views in the UI layer.

17. Build a contact form and implement the functionality to send form submissions to your email.
18. Create and utilize custom properties to show a summary of information about scheduled classes and to display the students full name.
19. Implement image upload functionality for the student create and edit forms.
20. Implement soft deletes for students, courses, and scheduled classes.
21. Utilize a session variable in the course workflow to define where the user should be sent (list of active or retired courses) when they click on the cancel button for edit or back on details.
22. Utilize session variables to define where a user should be sent after a soft delete.
23. Utilize a session variable to define the value of the status of a scheduled class status shown in the delete view.
24. Add functionality to allow the user to view the index of the Student Status controller in a table layout or tiled layout.
25. Create documentation for the project and link to the documentation from the Home/Index of the project.
26. Create a backup script of the database.
27. Ensure all code is pushed to the Github repo. (Note: ensure that you are not pushing usernames or passwords)
28. Each team member should create a live database and subdomain on SmarterAsp. They should execute the backup script on the live database, deploy the project files, and link to the live project from their personal site. The description of this project should include details on that person's individual contributions.