1. The project history is tracked using Git and the codebase is stored in a public repo on GitHub. Ensure the appsettings.json file is listed in the .gitignore so that it is not being tracked or committed.
2. Create the UI layer of your project.
3. Implement authentication using Identity Samples.
4. Implement Identity Users and Roles Management (IdentityAddons).
5. Select a multi-page template, download the files, and bring them into an _Archive folder in your UI project.
6. Create the data layer and utilize Entity Framework to create domain models for your database objects.
7. Update the connection strings in appsettings.json
8. 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.
9. Rename the _Layout to _OriginalLayout and then utilize the appropriate HTML file from the template to create a new _Layout
0. Update all navigation links and image file paths.
1. Build metadata (buddy) classes in the data layer.
2. Scaffold all Controllers & Views in the UI layer.
3. Build a contact form and implement the functionality to send form submissions to your email. Ensure that you are storing usernames and passwords in the appsettings.json and not hard coding them in the controller.
4. Style each view so that the application has a consistent look and feel.
5. Replace all Lorem Ipsum and scaffolded demo content.
6. Replace all template images with your photos or free stock images sourced from any of the resources provided in FUI.

FINAL CHECK POINT:

1. Implement client-side or server-side filters and paging for list views.
2. Implement a session-based shopping cart.
3. Utilize AJAX for CRUD functionality on the views for a lookup table like Categories.
4. Implement custom error handling in the project.
5. Add the image upload utility to the application to increate the maintainability of user uploads and increase performance of the application when displaying photos in a list view by leveraging thumbnails.
6. Push the final version of your code to your GitHub repo.
7. Link to the GitHub repo from your Personal Site.

HomeController	Index	Contact
Anonymous	x	x
Scheduling	x	x
Admin	x	x

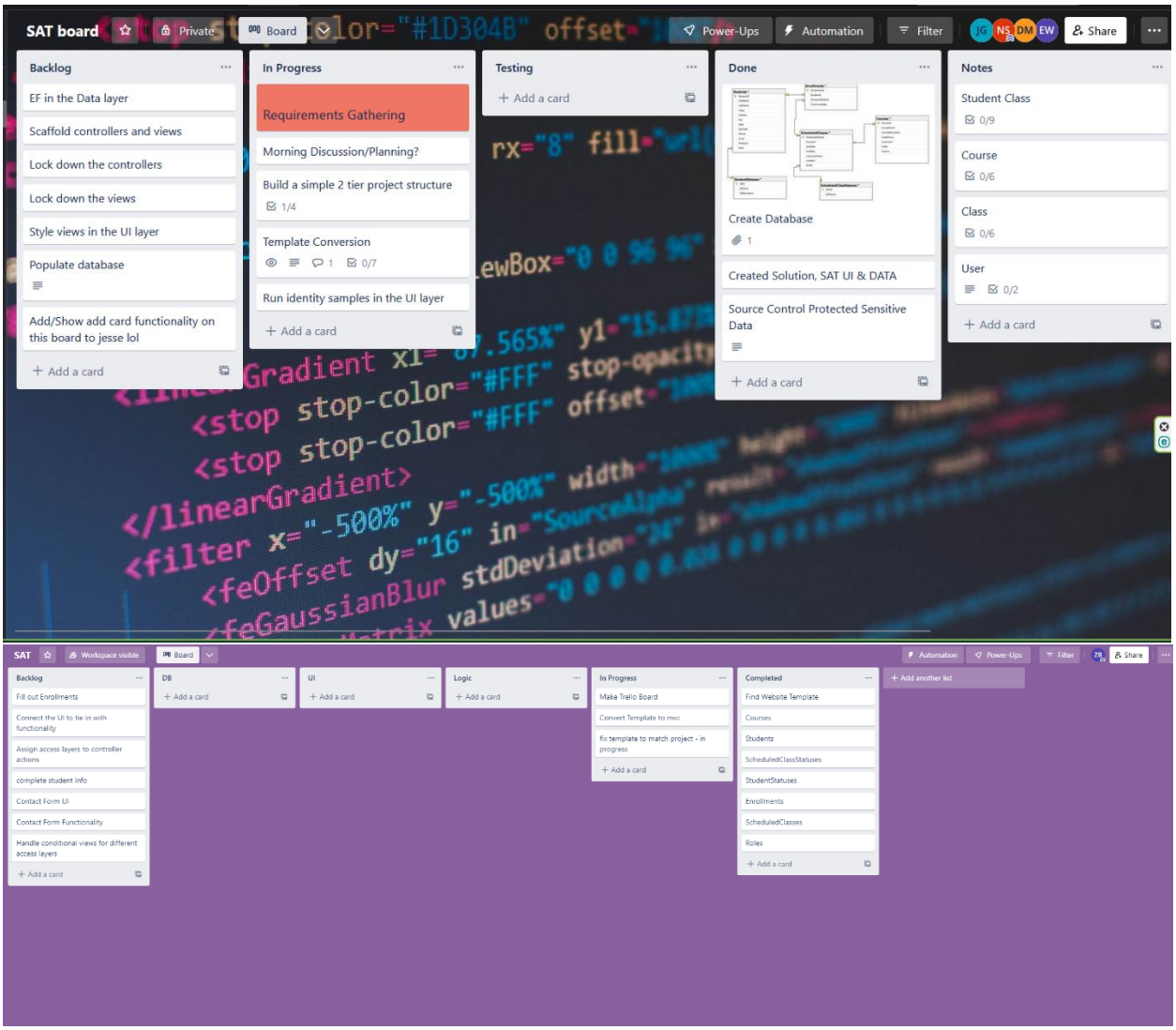
Courses	Active	Retired	Detail	Create	Edit	Delete
Anonymous						
Scheduling						
Admin	x	x	x	x	x	x

Enrollments	Index	Details	Create	Edit	Delete
Anonymous					
Scheduling	x	x	x	x	x
Admin	x	x	x	x	x

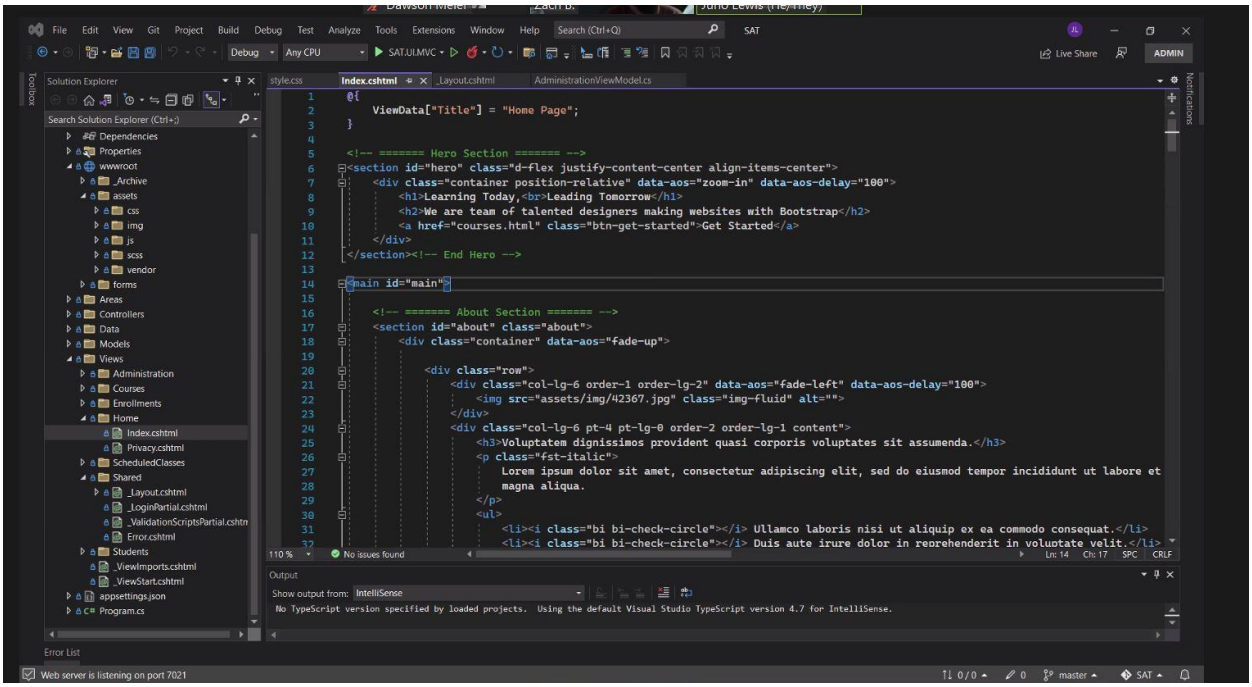
Scheduled Classes	Index	Details	Create	Edit	Delete
Anonymous					
Scheduling	x	x	x	x	
Admin	x	x	x	x	x

Students	Index	Details	Create	Edit	Delete
Anonymous					
Scheduling	x	x			
Admin	x	x	x	x	x

Student Statuses	Index	Details	Create	Edit	Delete
Anonymous					
Scheduling					
Admin	x	x	x	x	x


SAT board
Private
Board
Power-Ups
Automation
Filter
Share
Backlog
EF in the Data layer
Scaffold controllers and views
Lock down the controllers
Lock down the views
Style views in the UI layer
Populate database
Add/Show add card functionality on this board to jesse lol
Add a card
In Progress
Requirements Gathering
Morning Discussion/Planning?
Build a simple 2 tier project structure
1/4
Template Conversion
0/7
Run identity samples in the UI layer
Add a card
Testing
Add a card
Done
Create Database
1
Created Solution, SAT UI & DATA
Source Control Protected Sensitive Data
Add a card
Notes
Student Class
0/9
Course
0/6
Class
0/6
User
0/2
Add a card

SAT
Workspace visible
Board
Automation
Power-Ups
Filter
Share
Backlog
Fill out Enrollments
Connect the UI to tie in with functionality
Assign access layers to controller actions
complete student info
Contact Form UI
Contact Form Functionality
Handle conditional views for different access layers
Add a card
DB
Add a card
UI
Add a card
Logic
Add a card
In Progress
Make Trello Board
Convert Template to mvc
fix template to match project - in progress
Add a card
Completed
Find Website Template
Courses
Students
ScheduledClassStatuses
StudentStatuses
Enrollments
ScheduledClasses
Roles
Add a card
Add another list


File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Search (Ctrl+Q)
SAT
Debug
Any CPU
SAT.UI.MVC
Live Share
ADMIN
Solution Explorer
Search Solution Explorer (Ctrl+;)
Dependencies
Properties
wwwroot
_Archive
assets
css
img
js
scss
vendor
forms
Areas
Controllers
Data
Models
Views
Administration
Courses
Enrollments
Home
Index.cshtml
Privacy.cshtml
ScheduledClasses
Shared
_Layout.cshtml
_LoginPartial.cshtml
_ValidationScriptsPartial.cshtml
Error.cshtml
Students
_ViewImports.cshtml
_ViewStart.cshtml
appsettings.json
Program.cs
style.css
Index.cshtml
_Layout.cshtml
AdministrationViewModel.cs
@{
ViewData["Title"] = "Home Page";
}
<!-- ======= Hero Section ======= -->
<section id="hero" class="d-flex justify-content-center align-items-center">

<h1>Learning Today,
Leading Tomorrow</h1>
<h2>We are team of talented designers making websites with Bootstrap</h2>
Get Started

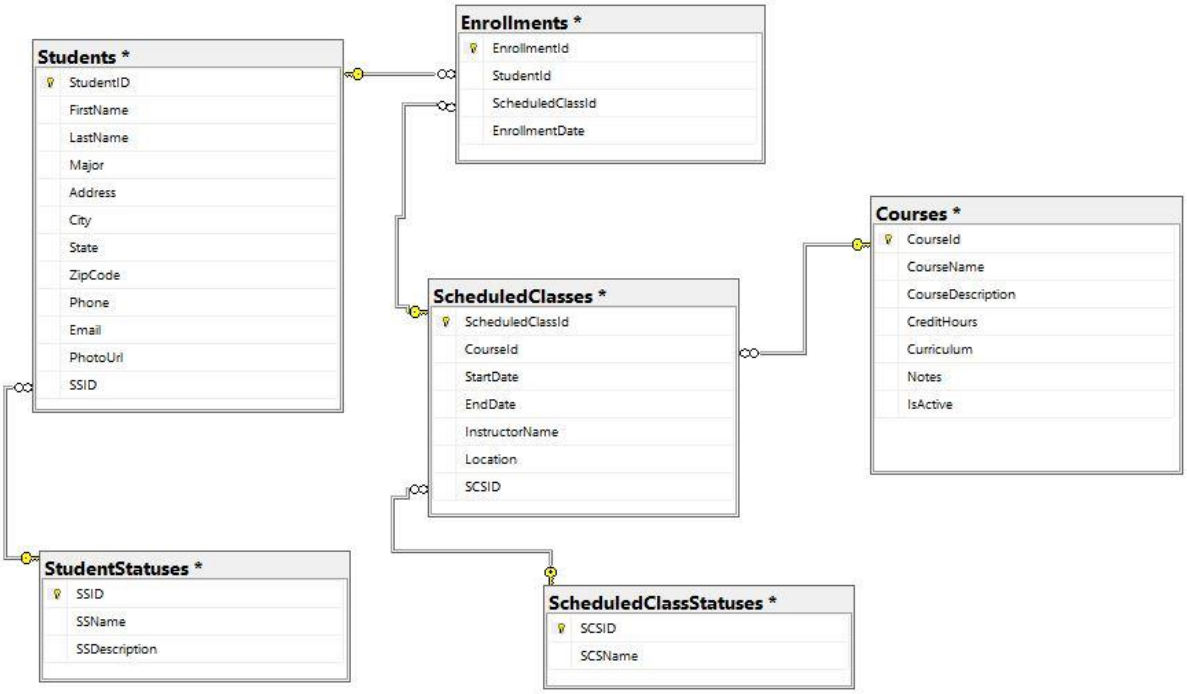
</section><!-- End Hero -->
<main id="main">
<!-- ======= About Section ======= -->
<section id="about" class="about">

<img src="assets/img/42367.jpg" class="img-fluid" alt="">

<h3>Voluptatem dignissimos provident quasi corporis voluptates sit assumenda.</h3>

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et
magna aliqua.

 Ullamco laboris nisi ut aliquip ex ea commodo consequat.
 Duis aute irure dolor in reprehenderit in voluptate velit.
110 %
No issues found
Ln: 14 Ch: 17 SPC CRLF
Output
Show output from: IntelliSense
No TypeScript version specified by loaded projects. Using the default Visual Studio TypeScript version 4.7 for IntelliSense.
Error List
Web server is listening on port 7021
master
SAT


Students *
StudentID
FirstName
LastName
Major
Address
City
State
ZipCode
Phone
Email
PhotoUrl
SSID
Enrollments *
EnrollmentId
StudentId
ScheduledClassId
EnrollmentDate
Courses *
CourseId
CourseName
CourseDescription
CreditHours
Curriculum
Notes
IsActive
ScheduledClasses *
ScheduledClassId
CourseId
StartDate
EndDate
InstructorName
Location
SCSID
StudentStatuses *
SSID
SSName
SSDescription
ScheduledClassStatuses *
SCSID
SCSName