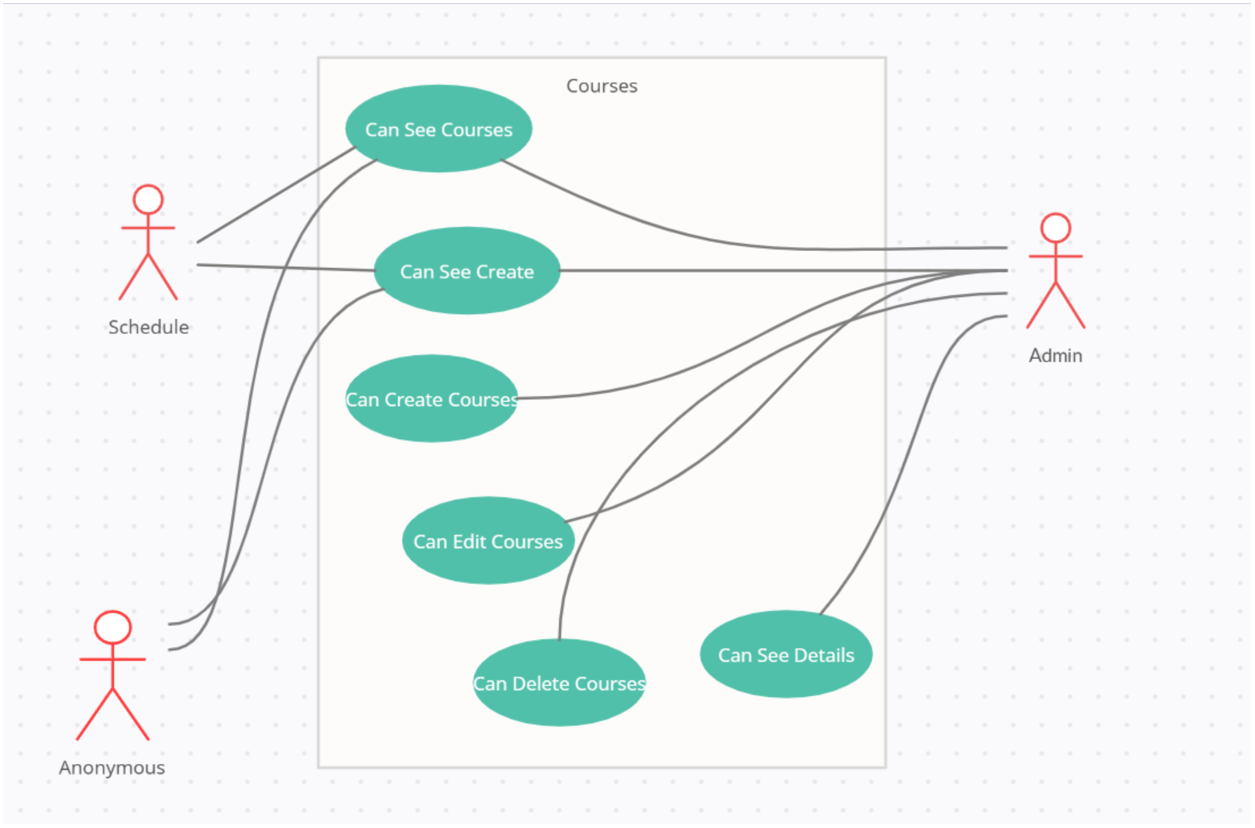
SAT Project Documentation

Requirement List

- A Courses Controller with CRUD views
- A Students Controller with CRUD views
- An Enrollment Controller with CRUD views
- A Scheduled Classes Controller with CRUD views
- Login Functionality implemented
- CRUD functionality to every page
- A Database holding all the data that is accessed from the website
- Using a Script to create mock data in our database
- Admin account
- Locking some CRUD functions behind Admin access
- A template converted into MVC and Customized
- Functioning Nav-Bar

Admin accounts are able to create courses, edit courses, delete courses and they can see details

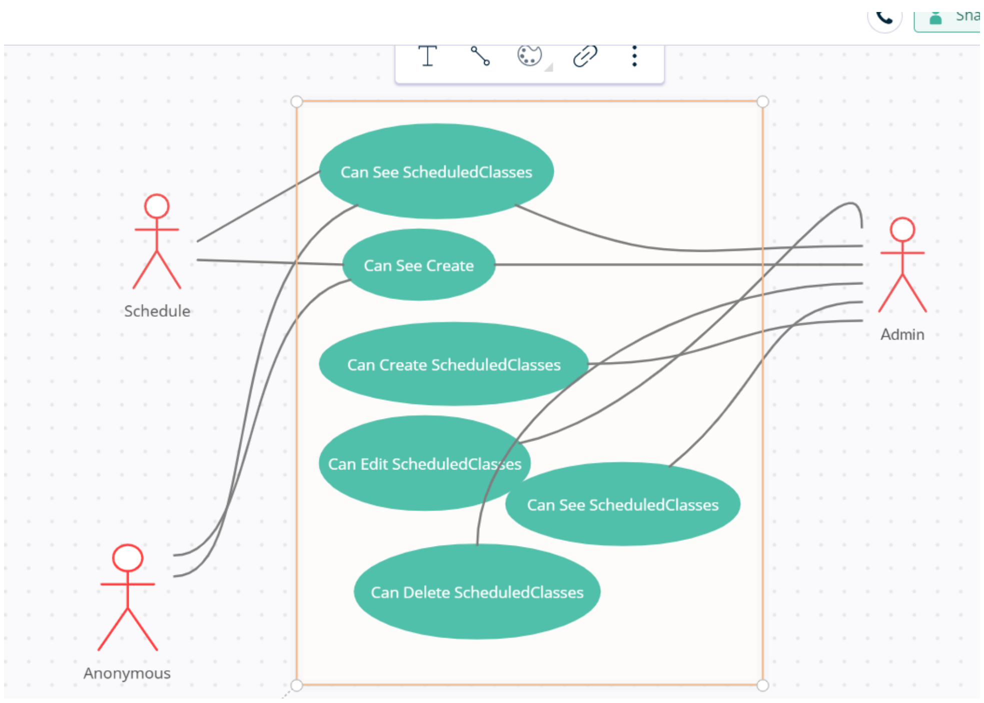
All accounts can see courses and can see the option to create courses



Admin accounts are able to create ScheduledClasses, edit ScheduledClasses, delete ScheduledClasses and they can see details

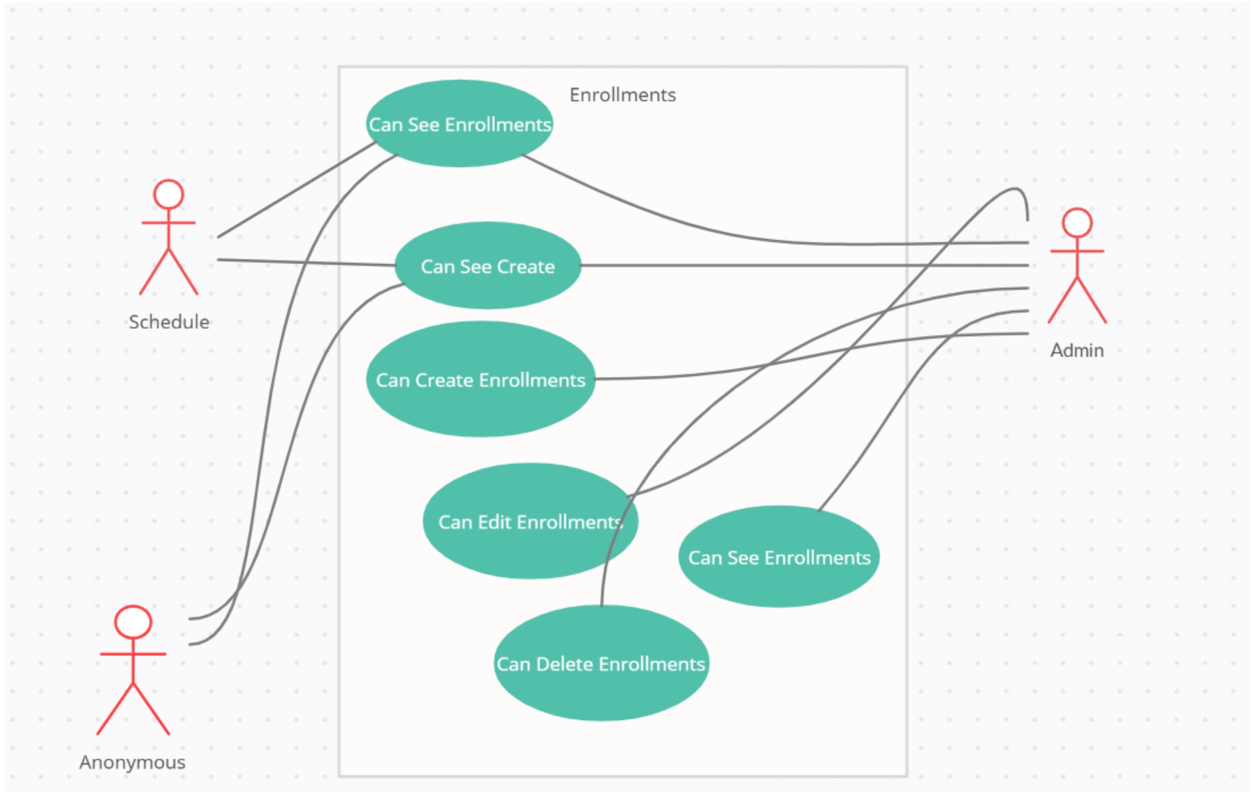
All accounts can see ScheduledClasses and can see the option to create ScheduledClasses

SAT Project Documentation



Admin accounts are able to create Enrollments, edit Enrollments, delete Enrollments and they can see details

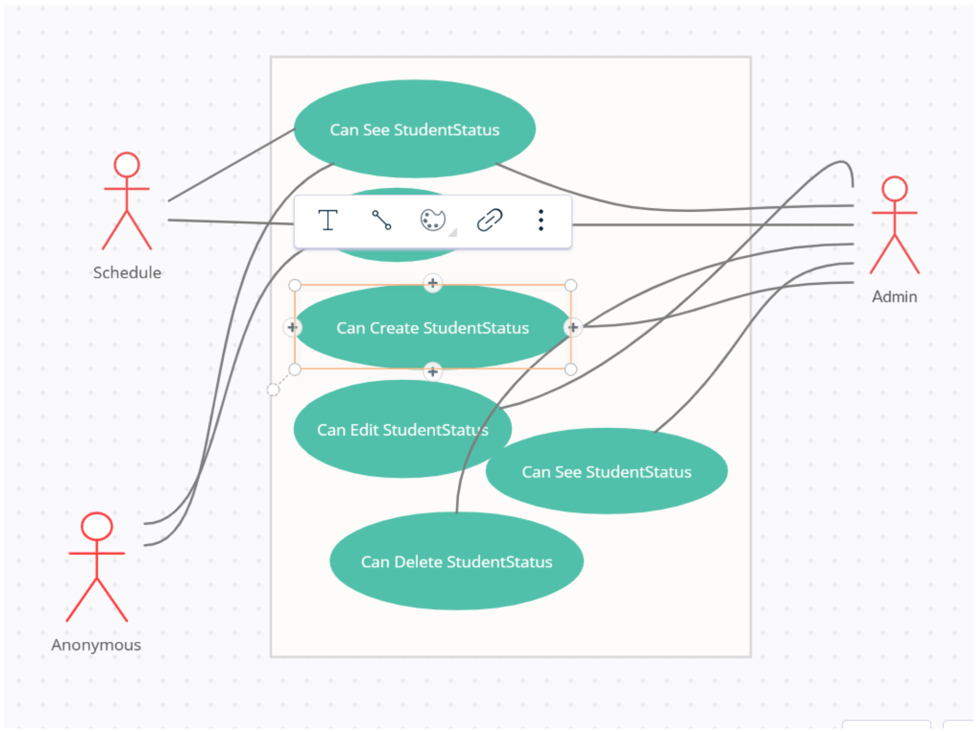
All accounts can see Enrollments and can see the option to create Enrollments



SAT Project Documentation

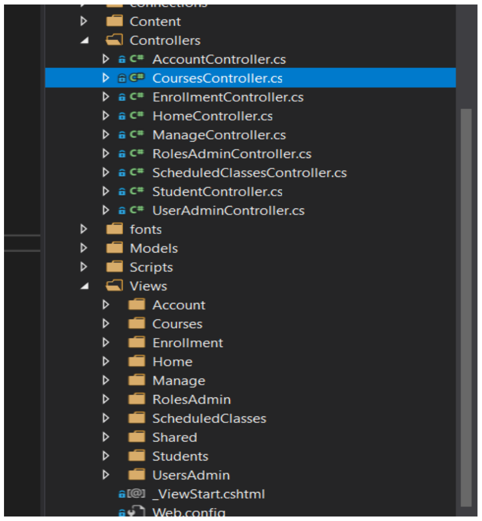
Admin accounts are able to create StudentStatus, edit StudentStatus, delete StudentStatus and they can see details

All accounts can see StudentStatus and can see the option to create StudentStatus



Requirement List

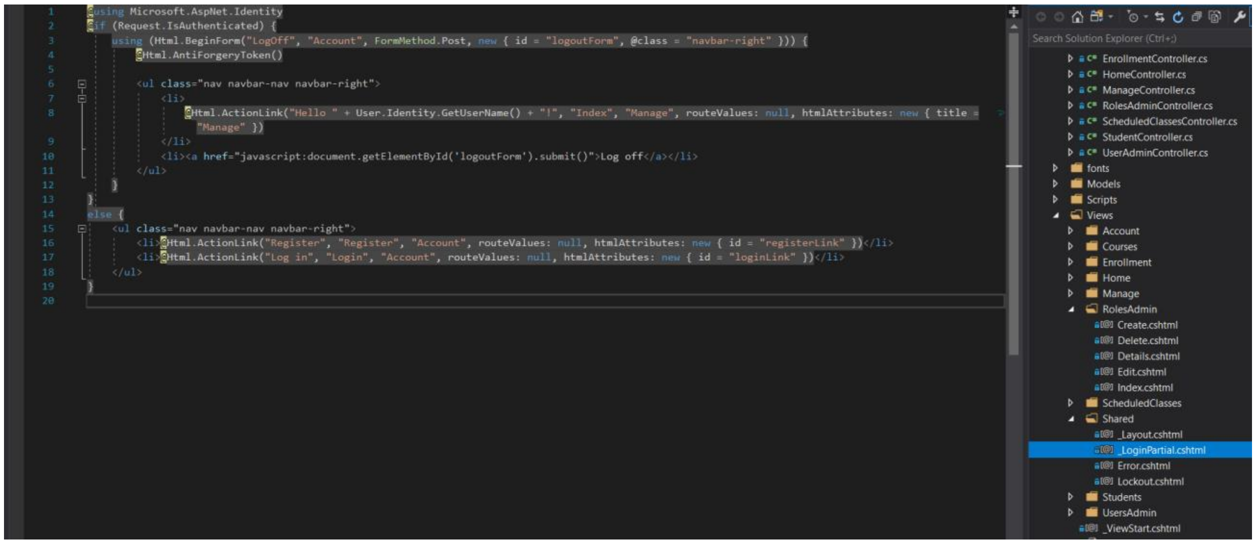
- A Courses Controller with CRUD views
- A Students Controller with CRUD views
- An Enrollment Controller with CRUD views
- A Scheduled Classes Controller with CRUD views



```
Content
Controllers
    AccountController.cs
    CoursesController.cs
    EnrollmentController.cs
    HomeController.cs
    ManageController.cs
    RolesAdminController.cs
    ScheduledClassesController.cs
    StudentController.cs
    UserAdminController.cs
fonts
Models
Scripts
Views
    Account
    Courses
    Enrollment
    Home
    Manage
    RolesAdmin
    ScheduledClasses
    Shared
    Students
    UsersAdmin
    _ViewStart.cshtml
    Web.config
```

SAT Project Documentation

- Login Functionality implemented

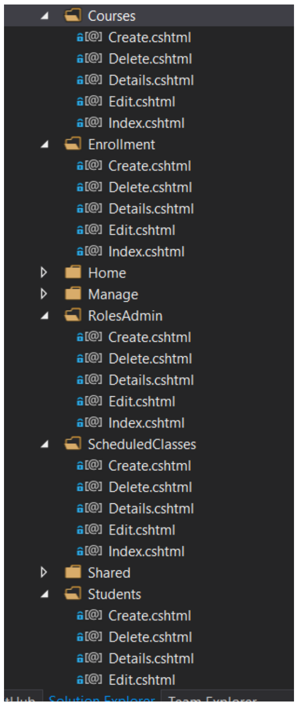


```
@using Microsoft.AspNet.Identity
@if (Request.IsAuthenticated) {
    using (Html.BeginForm("LogOff", "Account", FormMethod.Post, new { id = "logoutForm", @class = "navbar-right" })) {
        @Html.AntiForgeryToken()

        <ul class="nav navbar-nav navbar-right">
            <li>
                @Html.ActionLink("Hello " + User.Identity.GetUserName() + "!", "Index", "Manage", routeValues: null, htmlAttributes: new { title =
                  "Manage" })
            </li>
            <li><a href="javascript:document.getElementById('logoutForm').submit()">Log off</a></li>
        </ul>
    }
}
else {
    <ul class="nav navbar-nav navbar-right">
        <li>@Html.ActionLink("Register", "Register", "Account", routeValues: null, htmlAttributes: new { id = "registerLink" })</li>
        <li>@Html.ActionLink("Log in", "Login", "Account", routeValues: null, htmlAttributes: new { id = "loginLink" })</li>
    </ul>
}
```

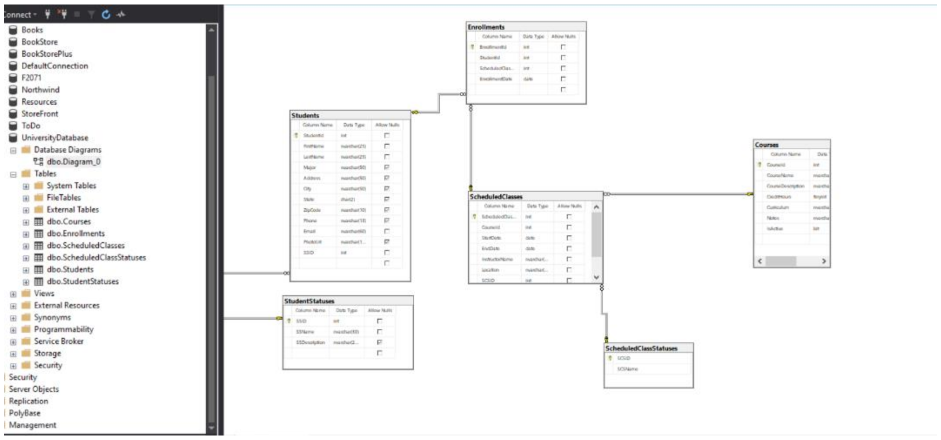
```
Search Solution Explorer (Ctrl+;)
    EnrollmentController.cs
    HomeController.cs
    ManageController.cs
    RolesAdminController.cs
    ScheduledClassesController.cs
    StudentController.cs
    UserAdminController.cs
  fonts
  Models
  Scripts
  Views
    Account
    Courses
    Enrollment
    Home
    Manage
    RolesAdmin
      Create.cshtml
      Delete.cshtml
      Details.cshtml
      Edit.cshtml
      Index.cshtml
    ScheduledClasses
    Shared
      _Layout.cshtml
      _LoginPartial.cshtml
      Error.cshtml
      Lockout.cshtml
    Students
    UsersAdmin
    _ViewStart.cshtml
```

- CRUD functionality to every page

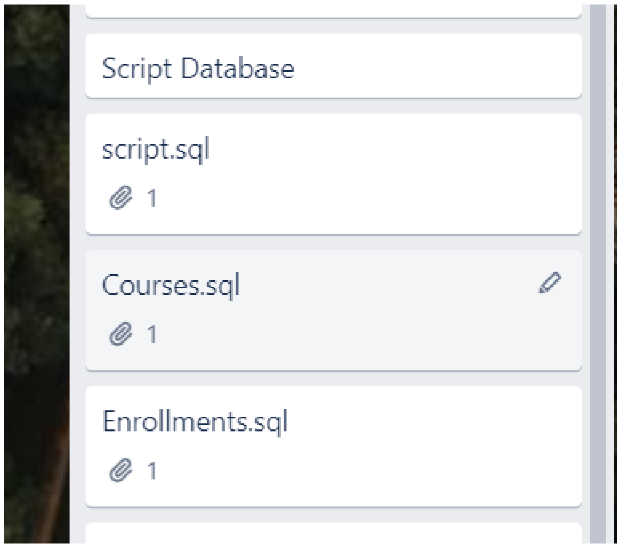


```
Courses
    Create.cshtml
    Delete.cshtml
    Details.cshtml
    Edit.cshtml
    Index.cshtml
Enrollment
    Create.cshtml
    Delete.cshtml
    Details.cshtml
    Edit.cshtml
    Index.cshtml
Home
Manage
RolesAdmin
    Create.cshtml
    Delete.cshtml
    Details.cshtml
    Edit.cshtml
    Index.cshtml
ScheduledClasses
    Create.cshtml
    Delete.cshtml
    Details.cshtml
    Edit.cshtml
    Index.cshtml
Shared
Students
    Create.cshtml
    Delete.cshtml
    Details.cshtml
    Edit.cshtml
```

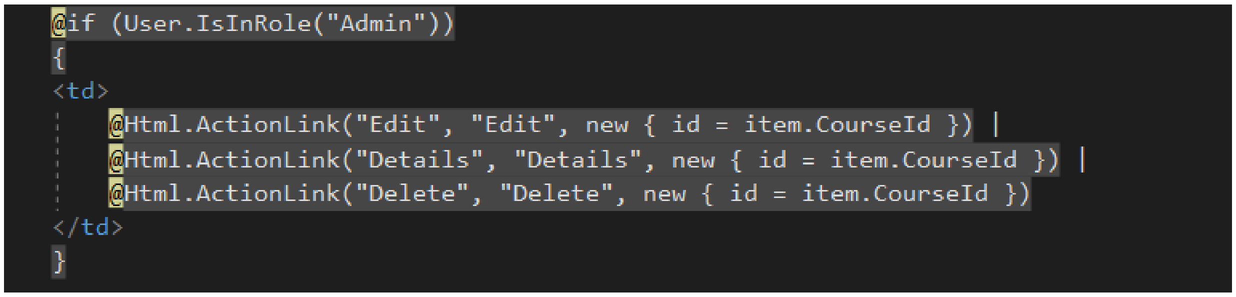
- A Database holding all the data that is accessed from the website



- Using a Script to create mock data in our database

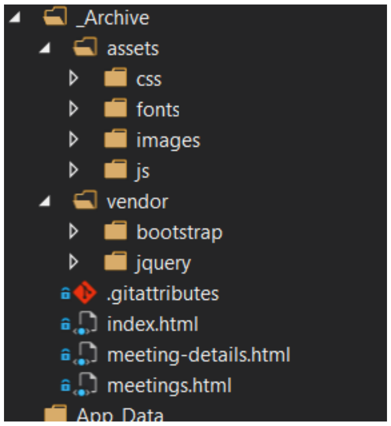


- Admin account
- Locking some CRUD functions behind Admin access



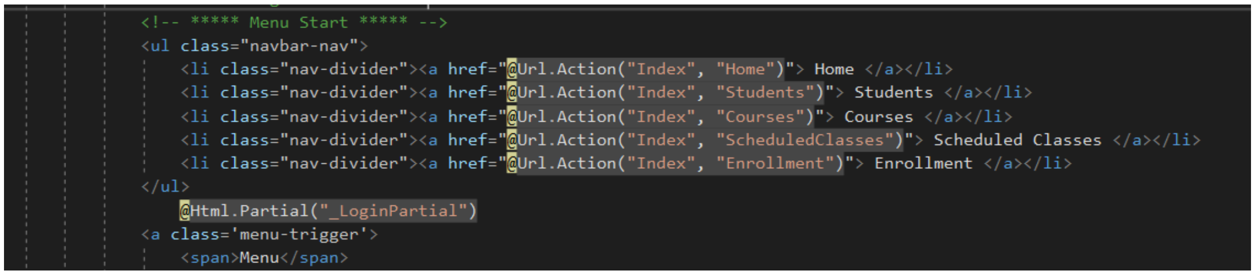
```
@if (User.IsInRole("Admin"))
{
<td>
    @Html.ActionLink("Edit", "Edit", new { id = item.CourseId }) |
    @Html.ActionLink("Details", "Details", new { id = item.CourseId }) |
    @Html.ActionLink("Delete", "Delete", new { id = item.CourseId })
</td>
}
```

- A template converted into MVC and Customized



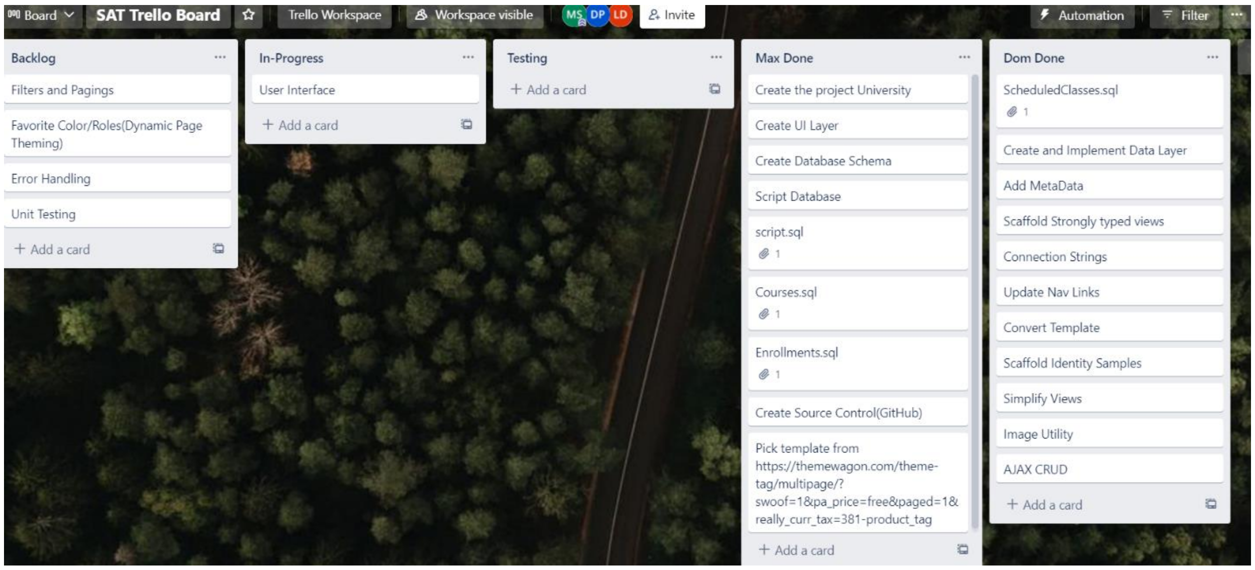
```
_Archive
  assets
    css
    fonts
    images
    js
  vendor
    bootstrap
    jquery
  .gitattributes
  index.html
  meeting-details.html
  meetings.html
App Data
```

- Functioning Nav-Bar



```
<!-- ***** Menu Start ***** -->
<ul class="navbar-nav">
    <li class="nav-divider"><a href="@Url.Action("Index", "Home")"> Home </a></li>
    <li class="nav-divider"><a href="@Url.Action("Index", "Students")"> Students </a></li>
    <li class="nav-divider"><a href="@Url.Action("Index", "Courses")"> Courses </a></li>
    <li class="nav-divider"><a href="@Url.Action("Index", "ScheduledClasses")"> Scheduled Classes </a></li>
    <li class="nav-divider"><a href="@Url.Action("Index", "Enrollment")"> Enrollment </a></li>
</ul>
    @Html.Partial("_LoginPartial")
<a class='menu-trigger'>
    <span>Menu</span>
```

Trello Board



Backlog
- Filters and Pagings
- Favorite Color/Roles(Dynamic Page Theming)
- Error Handling
- Unit Testing
- + Add a card

In-Progress
- User Interface
- + Add a card

Testing
- + Add a card

Max Done
- Create the project University
- Create UI Layer
- Create Database Schema
- Script Database
- script.sql 1
- Courses.sql 1
- Enrollments.sql 1
- Create Source Control(GitHub)
- Pick template from https://themewagon.com/theme-tag/multipage/?swoof=1&pa_price=free&paged=1&really_curr_tax=381-product_tag
- + Add a card

Dom Done
- ScheduledClasses.sql 1
- Create and Implement Data Layer
- Add MetaData
- Scaffold Strongly typed views
- Connection Strings
- Update Nav Links
- Convert Template
- Scaffold Identity Samples
- Simplify Views
- Image Utility
- AJAX CRUD
- + Add a card