Organizely

Front-End

Kaja Grabowska
@KajaGrabowska

Angela Mitchell
@amitchell05

Back-End

Andrea Pruitt
@DreaP

Maricar Walters
@mlwalters

Description

Organizely will help college students manage their coursework online. Students will be able to add and view their classes on their personal accounts. Students will also be able to mark class and assignment due dates on their calendars and receive reminders. To help with structuring study time, students will have access to a Focus Zone where students will be able to use a Pomodoro Timer to help maintain focus during studying, as well as be able to track their study sessions. For daily affirmations, students can view and save inspirational quotes to their accounts.

Features

- Create, Read, Edit, and Delete Classes
- Create, Read, Edit, and Delete Study Tasks and Assignments
- View Calendar (Daily, Weekly, Monthly Views)
- Signup and Login with JSON Web Token Authentication
- View, Save, and Delete Inspirational Quotes

Planning - User Stories

As a student, I want to:

- Create and login into my personal account
- Create, view, edit, and delete assignments, tasks, and courses
- Save and delete inspirational quotes
- View calendar with events (weekly and monthly views)

Planning - Database

- A students table that stores courses, tasks, assignments, to assigned users
- A courses table that stores courses with its name, times, dates, and teacher to its assigned user
- An assignments table that stores assignments with its due date, name and its associated course to its assigned user
- A student tasks table that stores tasks with its name, priority, and due date to its assigned user
- A quotes table that stores quotes with its content and author to its assigned user

Technology Stack

Front-End
- Angular 11
- TypeScript
- JavaScript
- HTML5
- CSS3
- Bootstrap 4

Back-End
- C#
- ASP.NET 5
- Entity Framework Core
- SQLite
- Identity
- Swagger UI

Demo

What We Learned

Front-End

- Deeper understanding of Bootstrap
- Persistent database (SQLite); connect Front-End to Back-End
- Sending GET, POST, PUT, and DELETE requests to the database API
- User authentication
- GIT and Github (branches, pull requests, code reviews, etc.)
- FullCalendar plugin
- Hide parts of the application for unauthenticated users
- Manipulate data w/helper functions
- Learned how to debug back-end

Back-End

- Building a RESTful API
- Code-First approach
- Handling HTTP GET, POST, PUT, and DELETE requests
- Using HTTP PUT and DELETE methods
- SQLite and SQLite Studio
- Linq queries and lambda expressions for EF Core and SQLite connection
- Server-side JWT Authentication
- Using Swagger UI for testing API
- Cross-Origin Resource Sharing (CORS) configuration
- GIT and Github (branches, pull requests, code reviews, etc.)

What's Next

- Focus Zone (Pomodoro Timer, Study Session Tracking, Music Playlist)
- User profile
- User Interface/Web Design
- Deploy to Azure
- XML Documentation for the API