CST-391: JavaScript Web Application Development

Milestone 4: Angular Application

Tanner Van Lith

# Video Link:

## Code Walk Through Part 1:

<https://www.loom.com/share/72875dd0c4fb41b2ae0ed2e9f6c5bfeb>

## Code Walk through Part 2:

<https://www.loom.com/share/f420c7c3b7bd47f1b98ed7d3be8005b4>

## Code Walk through Part 3:

<https://www.loom.com/share/4aea761a24cd40278867996c905c6199>

## Application Demo:

<https://www.loom.com/share/f690ca470132427da0208bb320acee3a>

# Introduction:

The application being developed in this milestone is used to allow a student to have direct access task in all their classes. This application is designed to allow students to check into due dates, and how many tasks they. The enterprise framework that will be utilized in this application will be JavaScript with Express.js.

# Functionality:

As a student, I want to be able to see the to-do list from school categorized by class, so each student can properly allocate the time to their scholastics.

# DB ER Diagram:

Diagram

Description automatically generated

# UI Sitemap:

Diagram

Description automatically generated

# Initial UI Wireframes:

# A screenshot of a computer Description automatically generated with medium confidence

# UML Classes:

|  |
| --- |
| taskLoader |
| -taskLoaderID: int  -class: string  -task: string  -dueDate: string  -userID: int |
| +settaskLoaderID()  +gettaskLoaderID(taskLoaderID: number)  +setclass()  +getclass(class: string)  +settask()  +gettask(task: string)  +setdueDate()  +getdueDate(dueDate: string)  +setuserID()  +getuserID(userID: int) |

|  |
| --- |
| userProfile |
| -userID: number  -name: string  -password: string  -email: string |
| +setuserID()  +getuserID(userID: number)  +setname()  +getname(name: string)  +setpassword()  +getpassword(password: string)  +setemail()  +getemail(email: string) |

# Risks:

A major risk in the development of this web application is the possibility of data attacks, which will isolate the student information in this application and try to extract it to use exploit its vulnerabilities. This will be avoided through redundancy and padding of the database.

The unknowns in this application can be best prepared for through making sure timelines of all tasks are reported prior to starting and communications must be fluid through all levels of hierarchy in a project. Project progress will be vigorously tracked to allow for the next phases in the project to be most prepared, as well as separating the developer and client languages in the project. As in any project, doing these things will lower stress and limit crunch time, which will ultimately increase the quality of the product being produced. And attacks can be avoided through redundancy and padding around the database.