**School Management System (SMS)**

**App**

* **Module 1 Admission Management**



**CMPS 350 Project Phase 1 – WebApp UI Design and Implementation**

|  |  |
| --- | --- |
| **Group Id:** | T-11 |
| **Group Members:** | Mohamed  Abdelmoniem |

**Grading Rubric - In the Functionality column please specify either: *Working (completed x%)*, *Not Working (completed x%)* or *Not done*.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Criteria | % | Functionality\* | Quality of the implementation | Grade |
| 1) Design the app Web UI and navigation (10%).  2) Implement the app Web UI and navigation using HTML, CSS and JavaScript (30%)  3) Implement the client-side data access repositories to read/write the app data from/to IndexedDB (40%) | | **80%** | ***Working 100%*** | - Add notes (can delete)  - Notes: Visible to parents: Visible  - All working |  |
|  | Login | 5 | ***Working 100%*** |  |  |
|  | Register applicant | 8 | ***Working 100%*** |  |  |
|  | Update applicant | 5 | ***Working 100%*** |  |  |
|  | Get applications (+ providing links to navigate to other functions) | 12 | ***Working 100%*** |  |  |
|  | Add application | 10 | ***Working 100%*** |  |  |
|  | Update application | 8 | ***Working 100%*** |  |  |
|  | View application | 6 | ***Working 100%*** |  |  |
|  | Withdraw application | 5 | ***Working 100%*** |  |  |
|  | Attachments (list, add, delete) | 9 | ***Working 100%*** | Attachments should be viewable even when the application is withdrawn. |  |
|  | Notes (list, add, delete) | 9 | ***Working 100%*** | Notes should be viewable even when the application is withdrawn.  This requirement is missing: The coordinator can decide whether the added/updated note can be visible to the applicants. By default, notes are visible only to the Principal. |  |
|  | Tests (list, add, delete) | 6 | ***Working 100%*** | Test should be viewable even when the application is withdrawn.  Test subject should be a dropdown populated from the database. |  |
|  | Applications summary report | 9 | ***Working 100%*** |  |  |
|  | Overall App design and navigation  Overall code quality | 8 | ***Working 100%*** | Excellent but with fee issues:   * Style sheets could be consolidated in fewer files while removing duplication. * In json files better to have the first letter of properties lowercase. Also, the date values are not meaningful. Date properties could be stored in a readable date format. * Too many logo images. * JS subfolder should be renamed js. * Throughout the app make variables start with a lowercase letter. * Add/Update Application should reuse the same html and JavaScript. * Add/Update Account should reuse the same html and JavaScript. * Provide back button when relevant (e.g., in notes, attachments and tests page). * Principle should be replaced with Principal everywhere in the app and documentation. * All dropdowns must be dynamically filled with data from the database. * See more comments added to the testing screenshots. |  |
| **4) Application Design:** Entities Class Diagram and the Repositories Class diagram. | | **8** | ***Done*** | Excellent, few issues see comments below. | 7 |
| 5) Create test data JSON files for your module entities | | 6 | ***Working 100%*** | Excellent | 6 |
| **6) Testing documentation** using screen shots illustrating the testing results.  - Discussion of the project contribution of each team member. | | 6 |  |  |  |
| **Total** | | 100 |  | Outstanding work | 95 |

**\* Possible grading for functionality** - ***Working*** (get 60% of the assigned grade), ***Not*** ***working*** (lose 40% of assigned grade and ***Not done*** (get 0). The remaining grade is assigned to the quality of the implementation.

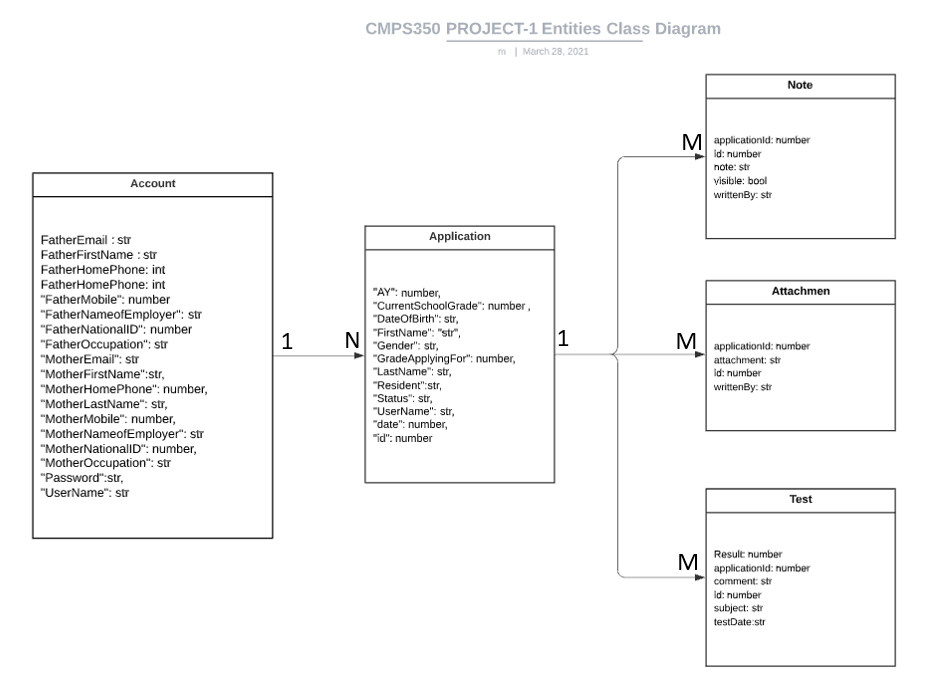
In case your implementation is not working then 40% of the grade will be lost and the remaining 60% will be determined based on of the code quality and how close your solution to the working implementation.

Solution quality also includes meaningful naming of identifiers (according to Android naming conventions), no redundant code, simple and efficient design, clean implementation without unnecessary files/code, use of comments where necessary, proper code formatting and indentation.

**Marks will be reduced** for **code duplication**, poor/inefficient coding practices, poor naming of identifiers, unclean/untidy submission, and unnecessary complex/poor user interface design.

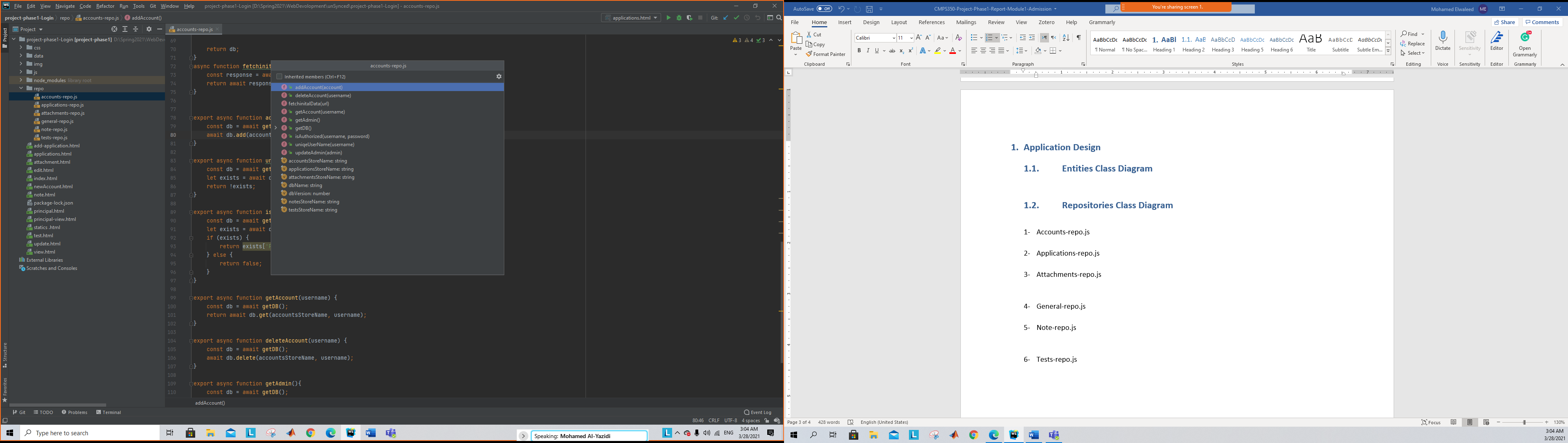
# Application Design

# Entities Class Diagram



# Repositories Class Diagram

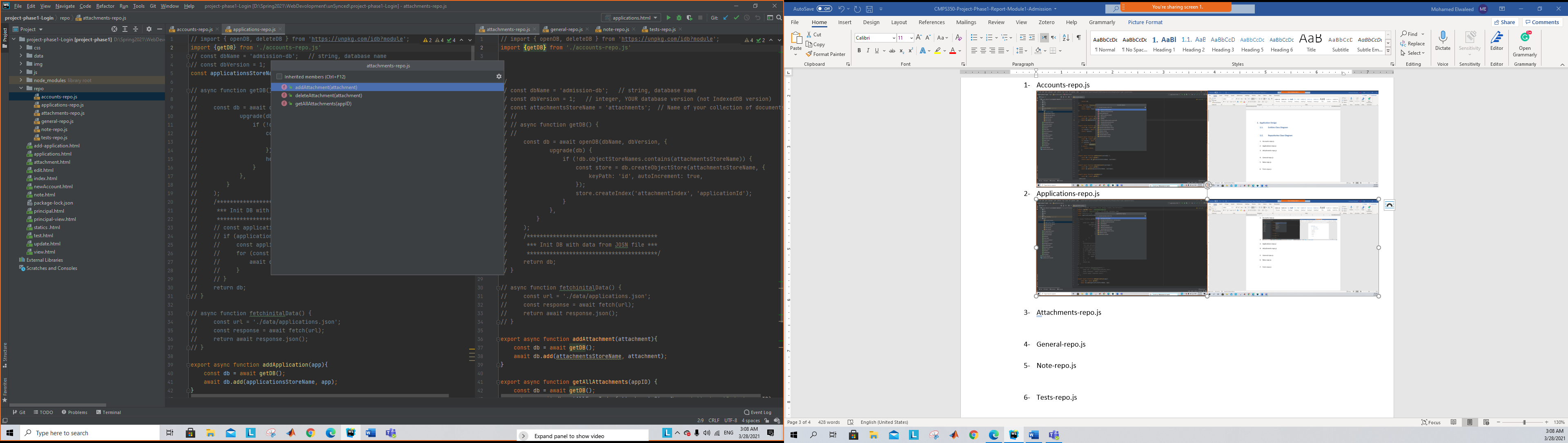
1. Accounts-repo.js



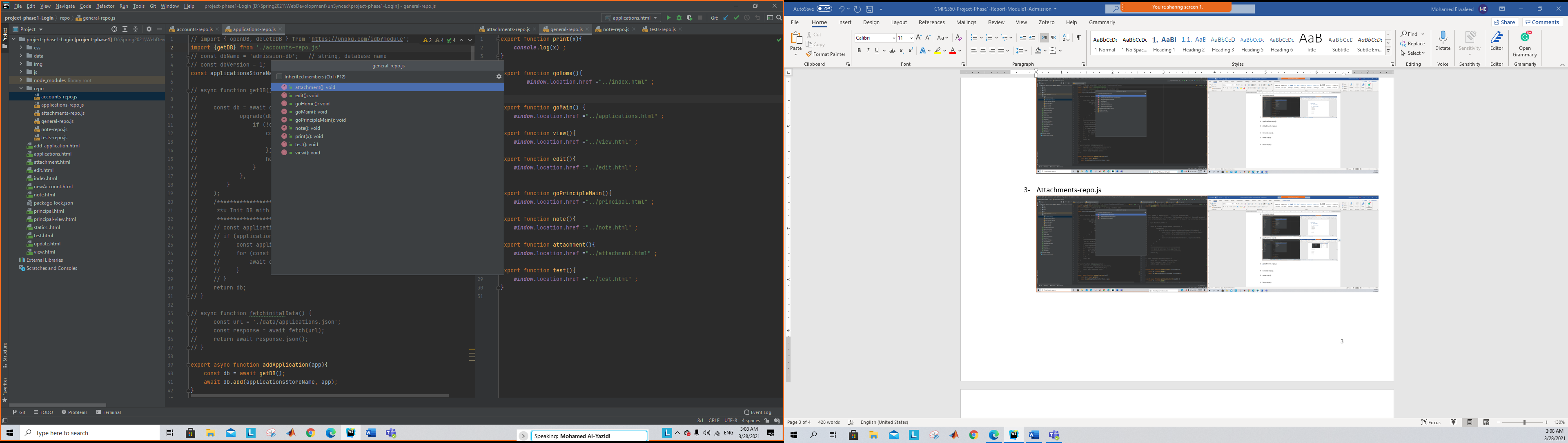
1. Applications-repo.js



1. Attachments-repo.js



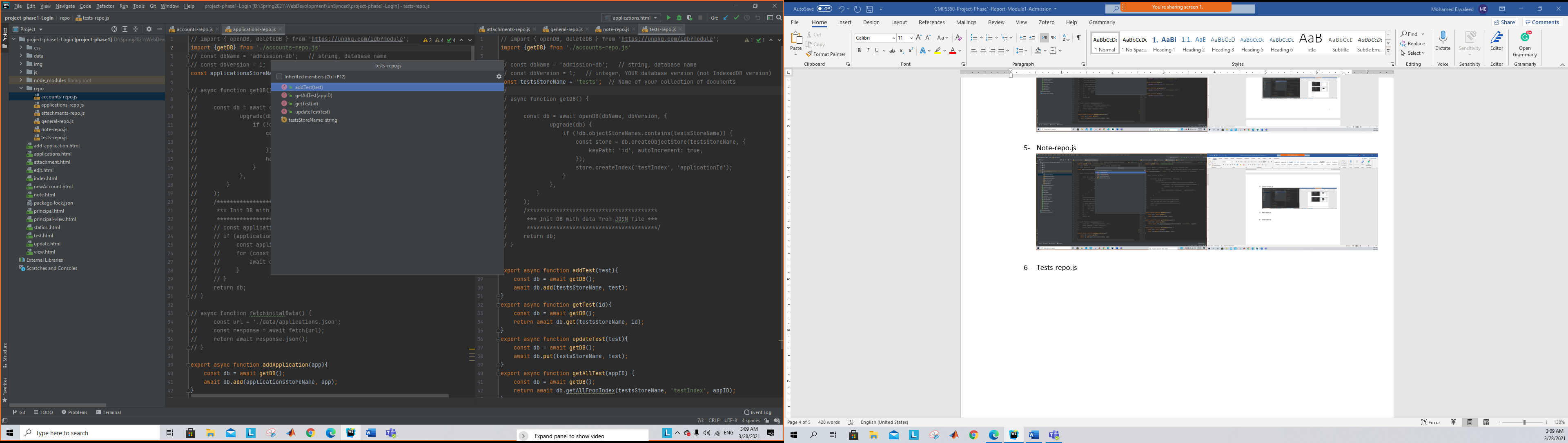
1. General-repo.js



1. Note-repo.js



1. Tests-repo.js



# App Testing

# Graphical user interface, application Description automatically generatedLogin

# Graphical user interface, table Description automatically generatedRegister applicant

Graphical user interface, application

Description automatically generatedGraphical user interface, application, table

Description automatically generated

# Graphical user interface, application Description automatically generatedGraphical user interface, application Description automatically generatedGraphical user interface, application Description automatically generatedUpdate applicant

# Get applications (+ providing links to navigate other functions)

Graphical user interface, text

Description automatically generated

# Add application

Graphical user interface, application, email

Description automatically generated

# Update application

Graphical user interface, text, application, email

Description automatically generated

# View application

Graphical user interface

Description automatically generated

# Withdraw application

Graphical user interface, text, application

Description automatically generatedGraphical user interface, text, application

Description automatically generated with medium confidence

# Attachments (list, add, delete)

Graphical user interface

Description automatically generated

# Notes (list, add, delete)

Graphical user interface, application

Description automatically generated

# Applications summary report

Graphical user interface, table

Description automatically generated

# Admin (Principal):

Main :

Table

Description automatically generated

# when you click on any application this page will appear:

Graphical user interface, application

Description automatically generatedGraphical user interface, application

Description automatically generatedGraphical user interface, application, email, website

Description automatically generatedGraphical user interface

Description automatically generatedGraphical user interface, text, application

Description automatically generatedA picture containing text, screenshot, indoor

Description automatically generated

# Summary of team member contributions

We divided the work between us and when one of us is stuck we have a meeting in WebEx.