**School Management System (SMS)**

**App**

* **Module 4 – Incidents Management**



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

|  |  |
| --- | --- |
| **Group Id:** | G7 |
| **Group Members:** | Nada  Muntaha |

**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%** |  |  | **71** |
|  | Login | 5 | Working (5) | Good but authenticate method poorly done as it reads ALL users from DB. It should only return the user matching the username (and the password). | 4.5 |
|  | Get Incidents (+ providing links to navigate to other functions) | 15 | Working (15) | Search on the UI (NOT on the DB). This is not efficient at all. All filtering should be done on the DB and NOT read all and filter on the client. | 13 |
|  | Report incident | 10 | Working (10) | Good | 10 |
|  | Update Incident | 7 | Working (7) | Should reuse the same UI and the same code as Report incident.  - Using + icon for Edit is confusing  - After Submit should go back to the list of incidents. | 6 |
|  | View Incident (including notes, attachments, and penalty) | 10 | Working (10) | Good but parent view should include viewing attachments.  Attachments and notes should be 1 to many (with incidents having 1 to many attachments/notes). | 9 |
|  | Add Penalty | 8 | Working (8) | Ok but easier to have it in a separate page rather than with get incidents (to keep things simpler). | 9 |
|  | Attachments (List, Add, Delete) | 12 | Working (8) | An incident should have 1 to many attachments (not just 1). | 9 |
|  | Notes (List, Add, Delete) | 12 | Working (8) | An incident should have 1 to many notes (not just 1). | 8 |
|  | Incidents Summary Report | 12 | Working (12) | grade-options should NOT be hardcoded. Reading all then filtering on the client is NOT efficient. | 11 |
|  | Overall App design and navigation | 9 | Working (9) | Good work but avoid hardcoding data in the app. | 9 |
| **4) Application Design:** Entities Class Diagram and the Repositories Class diagram. | | **8** | 8 | Some issues see comments added to the diagram | 6 |
| 5) Create test data JSON files for your module entities | | 6 | 6 | Location and type should have id instead of index and it should start with 1.  Rename type to incidentType.  Add if id property to user json and put teachers and staff in it. Remove login.json. | 6 |
| **6) Testing documentation** using screen shots illustrating the testing results.  - Discussion of the project contribution of each team member. | | 6 | 6 | Excellent | 6 |
| **Total** | | 100 |  | Excellent work | **89** |

**\* 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

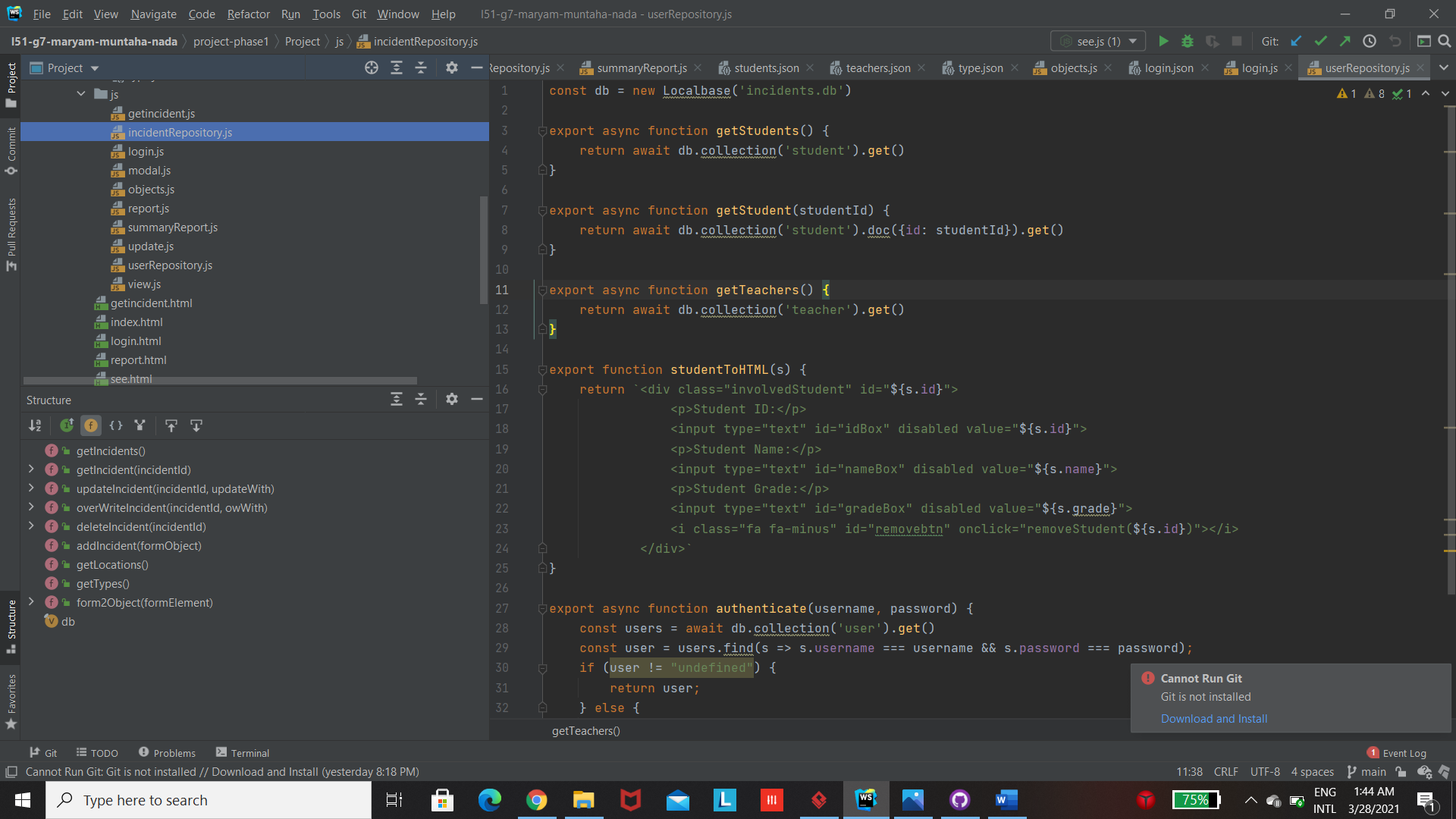
# Entities Class Diagram

Diagram

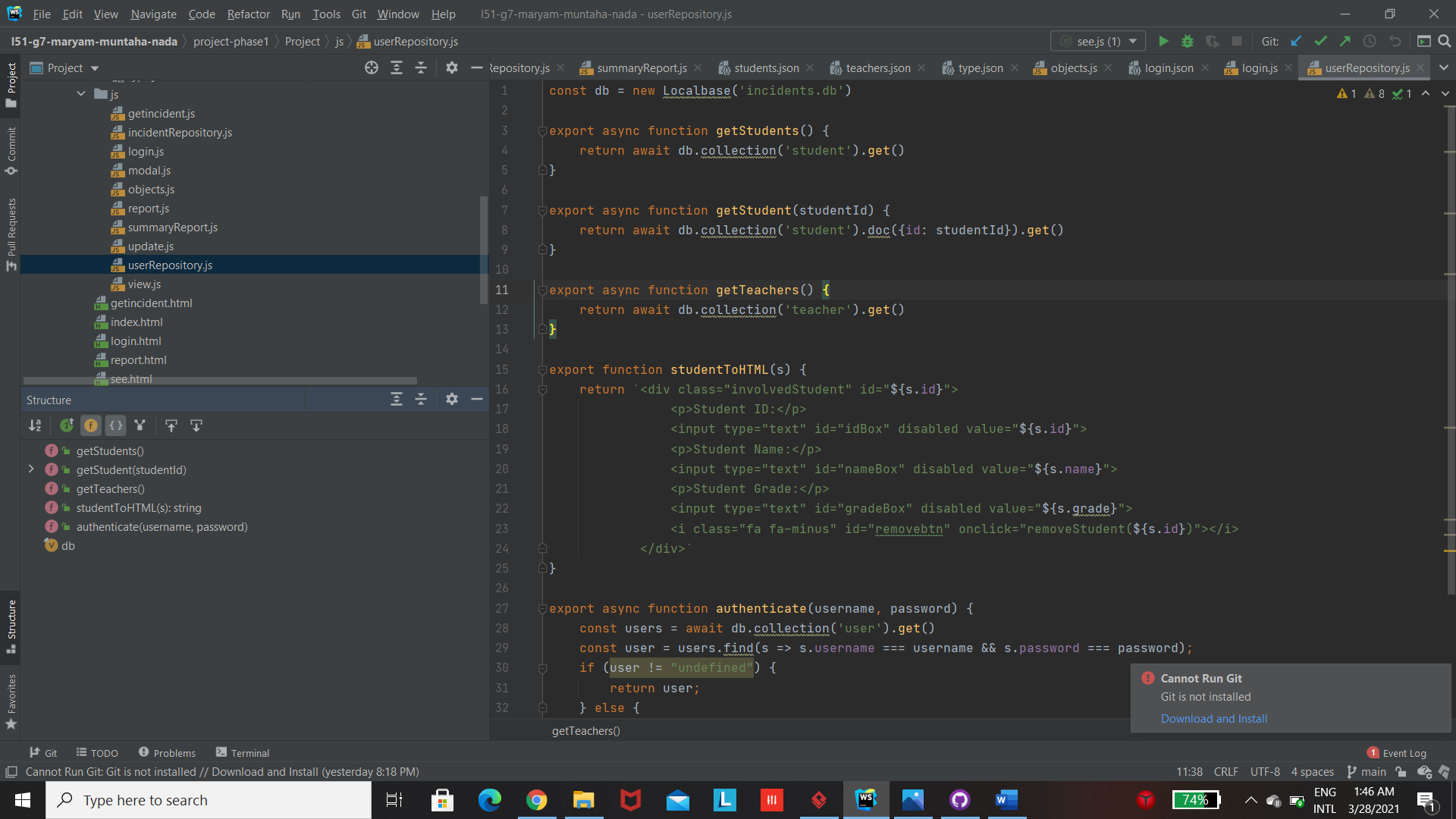
Description automatically generated



# Repositories Class Diagram

 **Repositories.js**

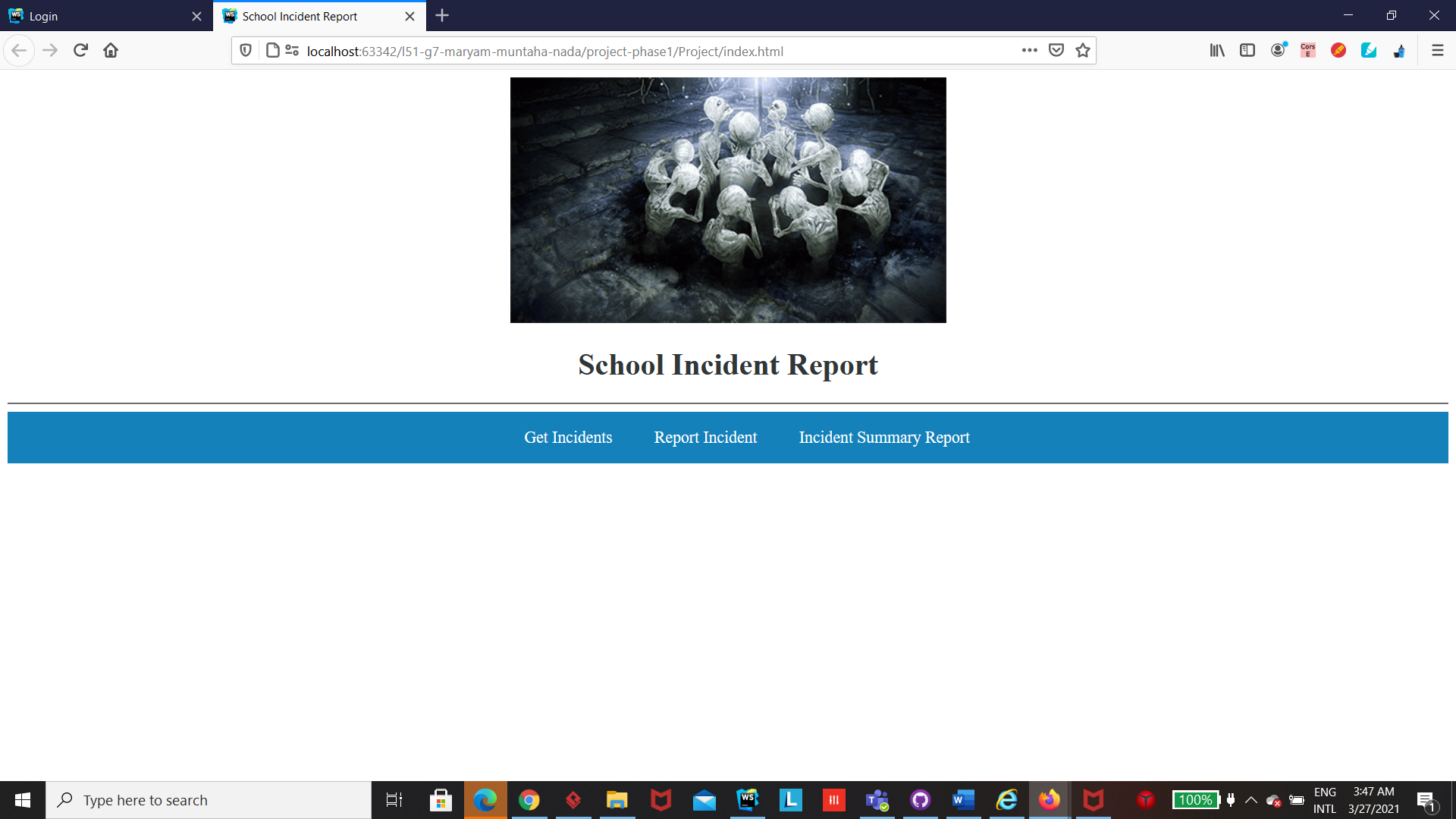


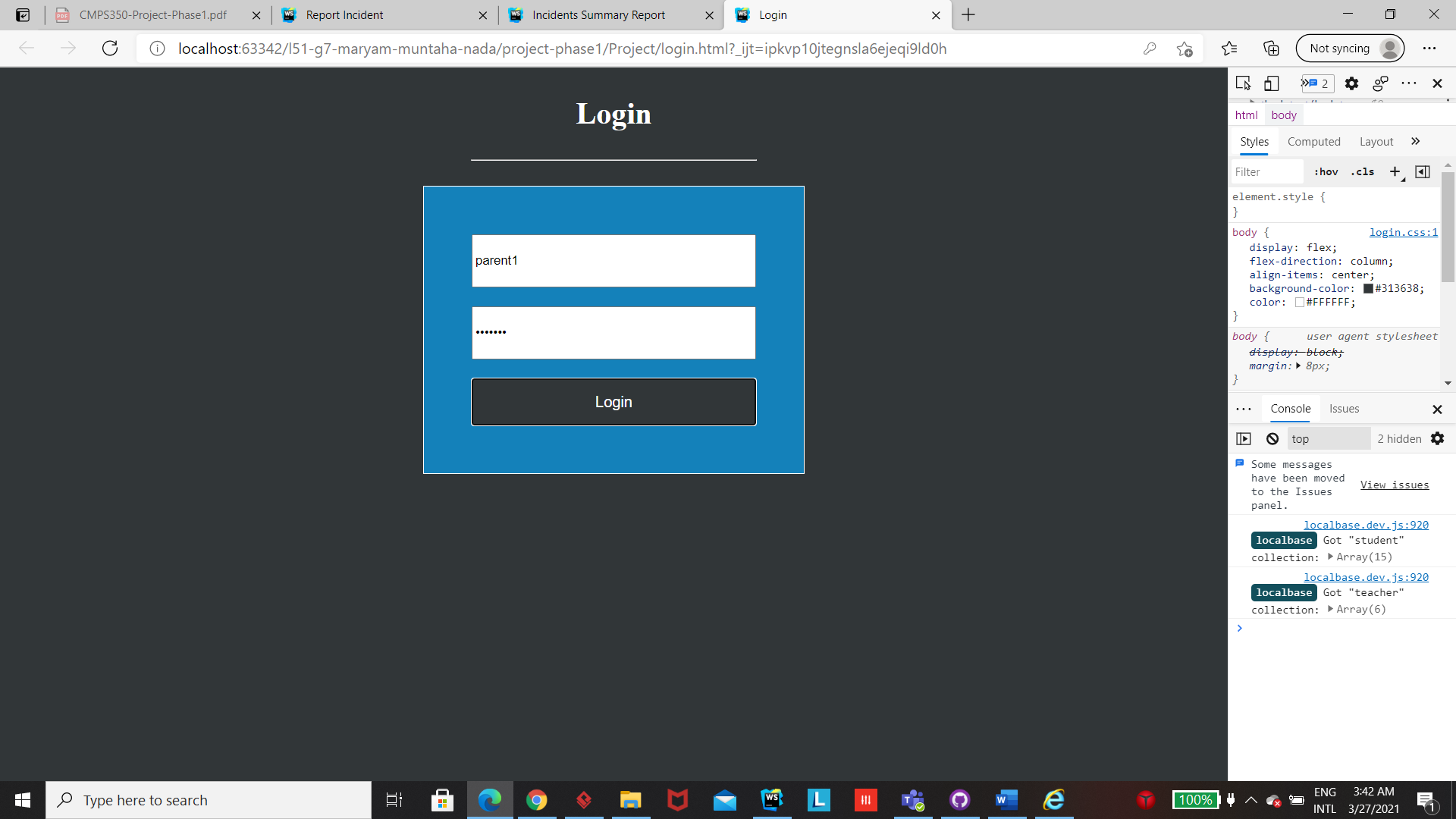


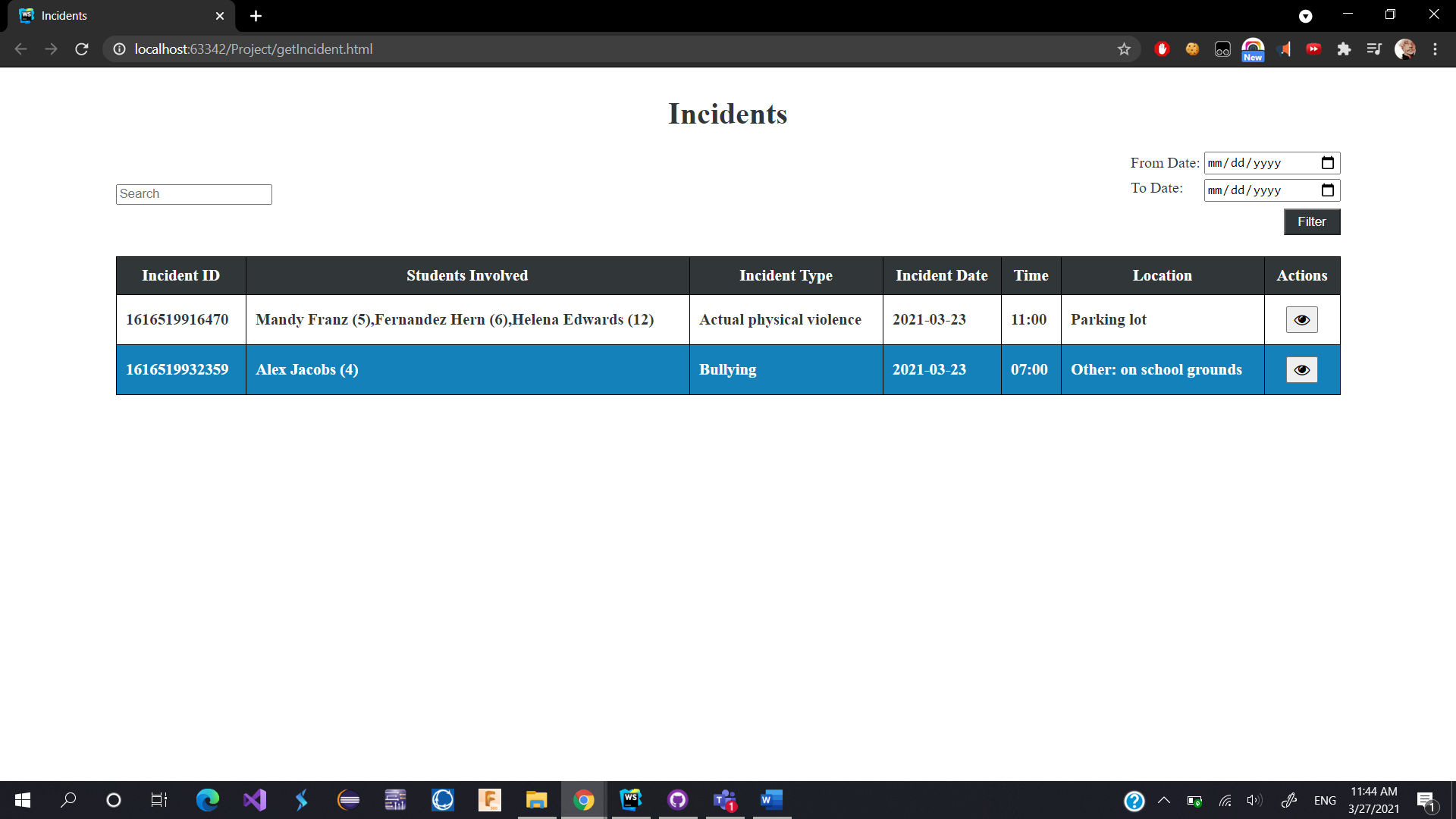
# App Testing

# Login

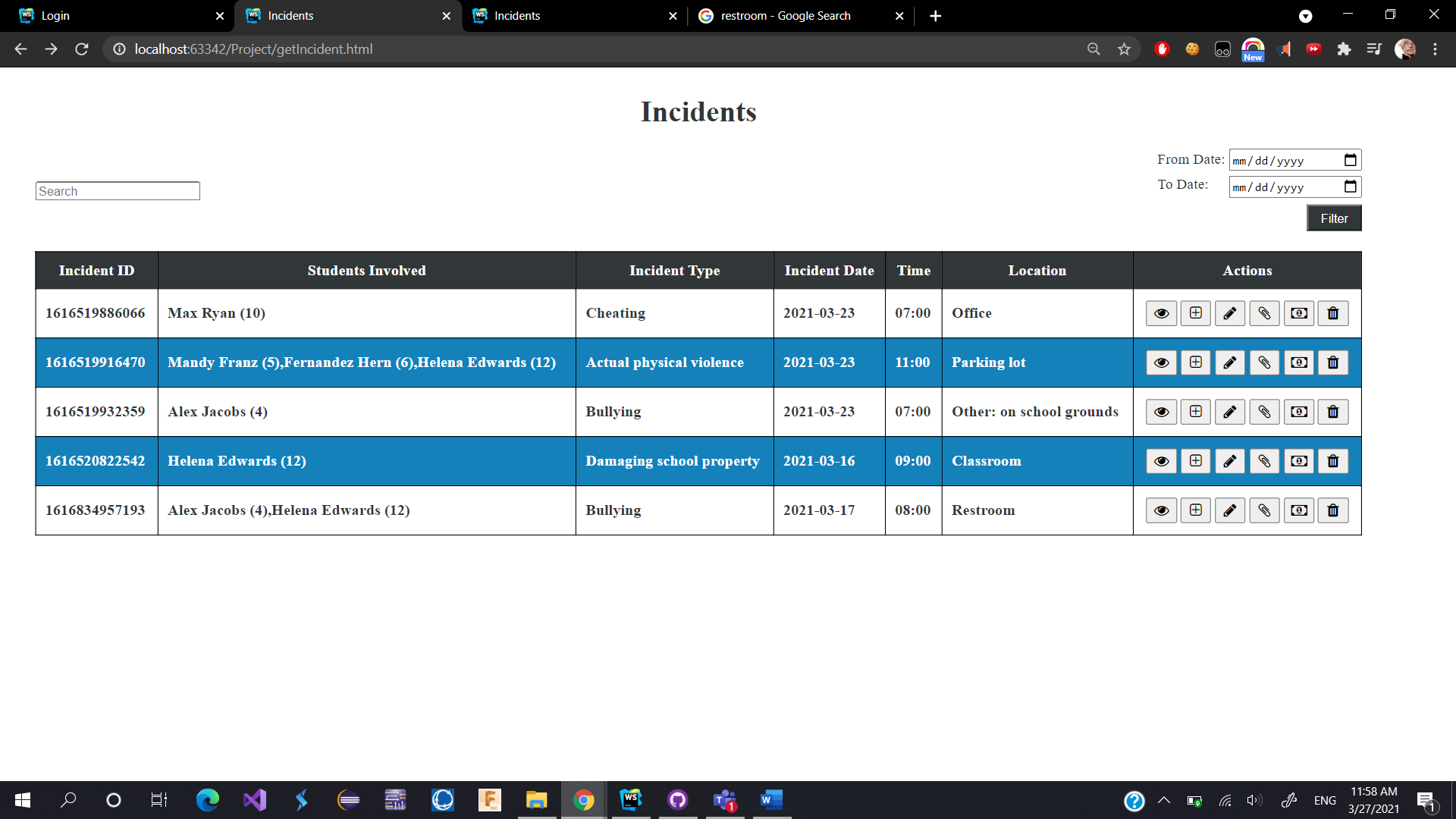
The staff Login page takes to the below home page

Parents login takes to the get incidents with only the parent’s children

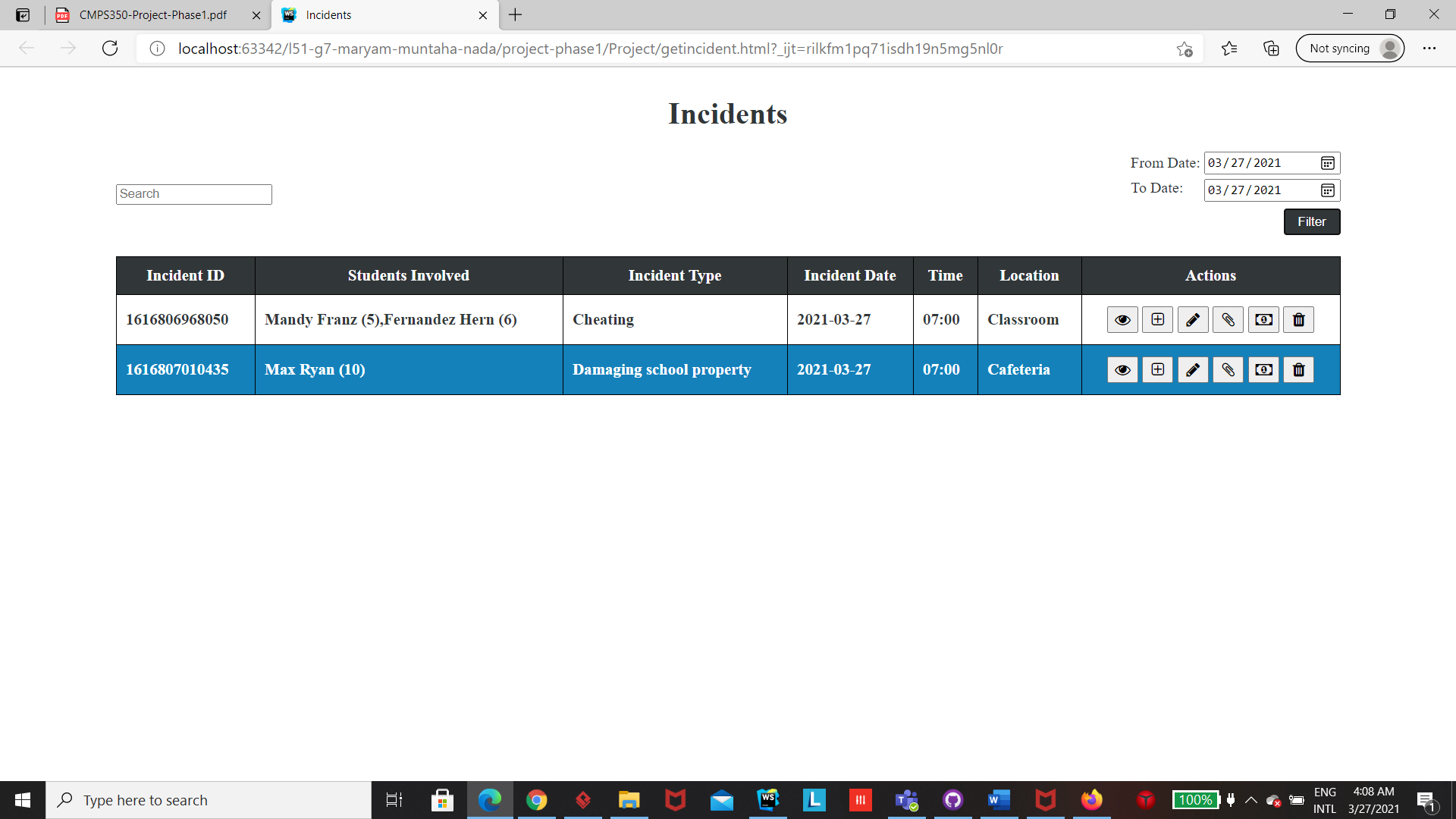




# Get Incidents (+ providing links to navigate to other functions)



Searching by date



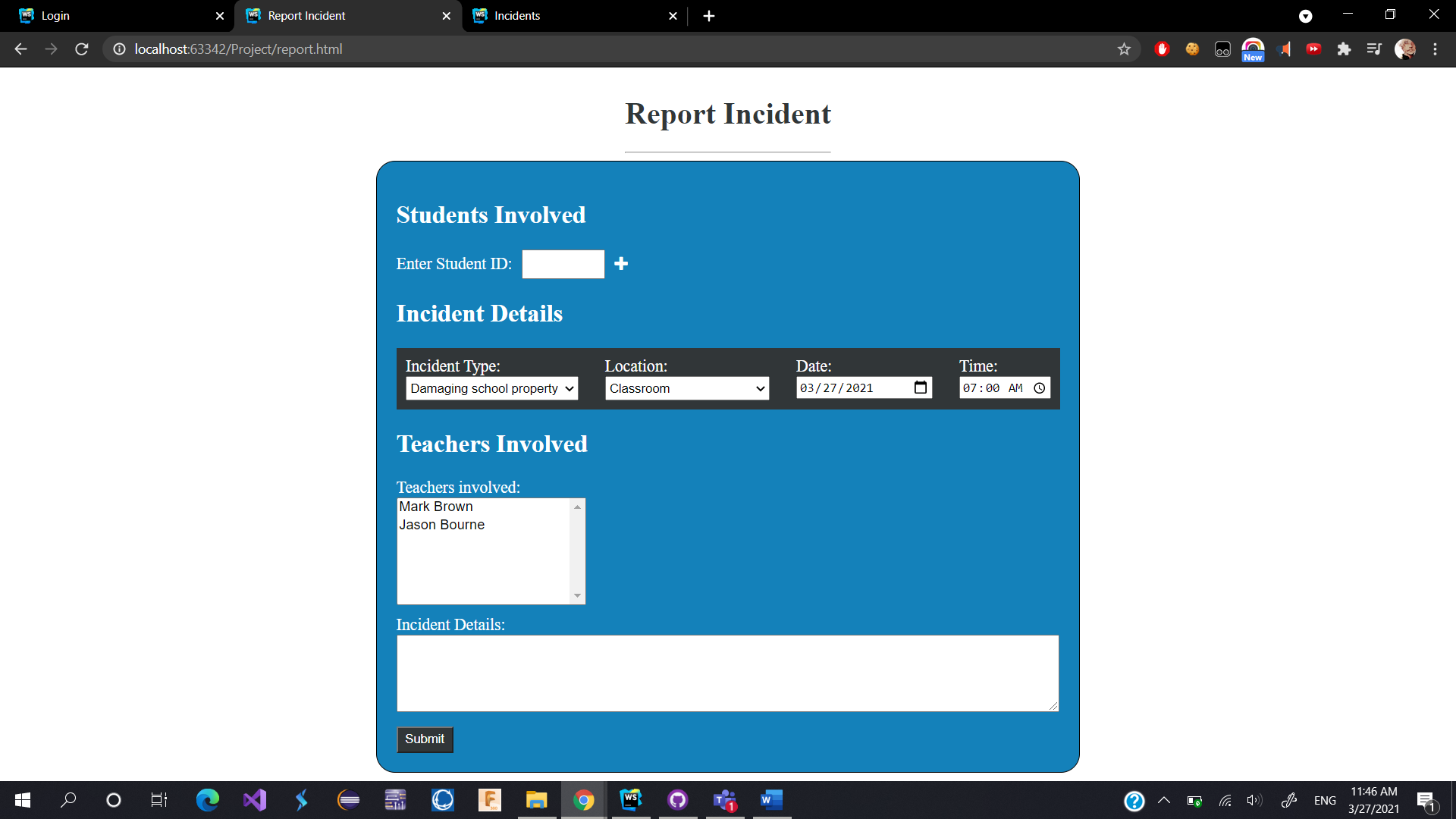
Searching by grade

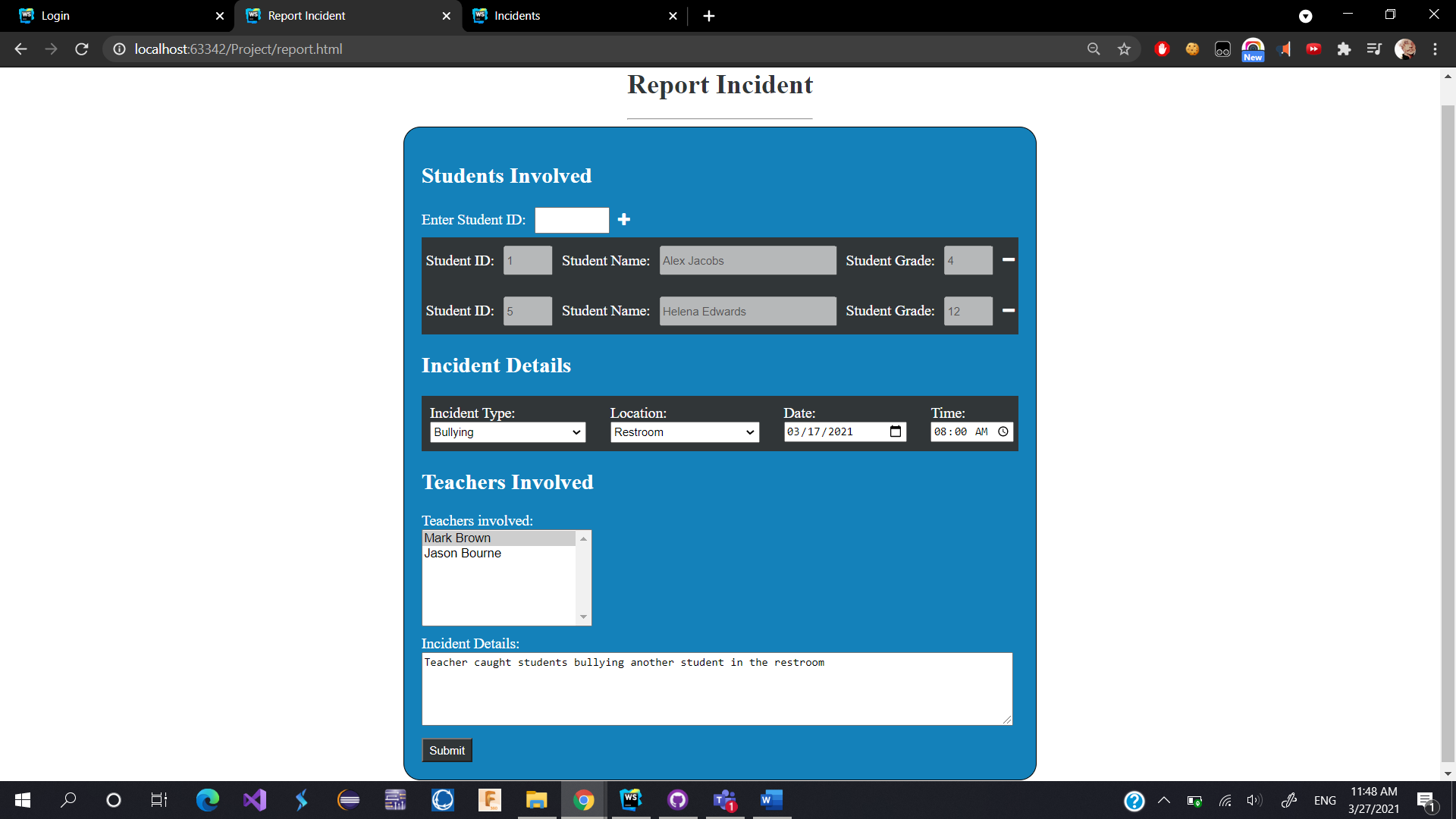
# 

Searching by student name

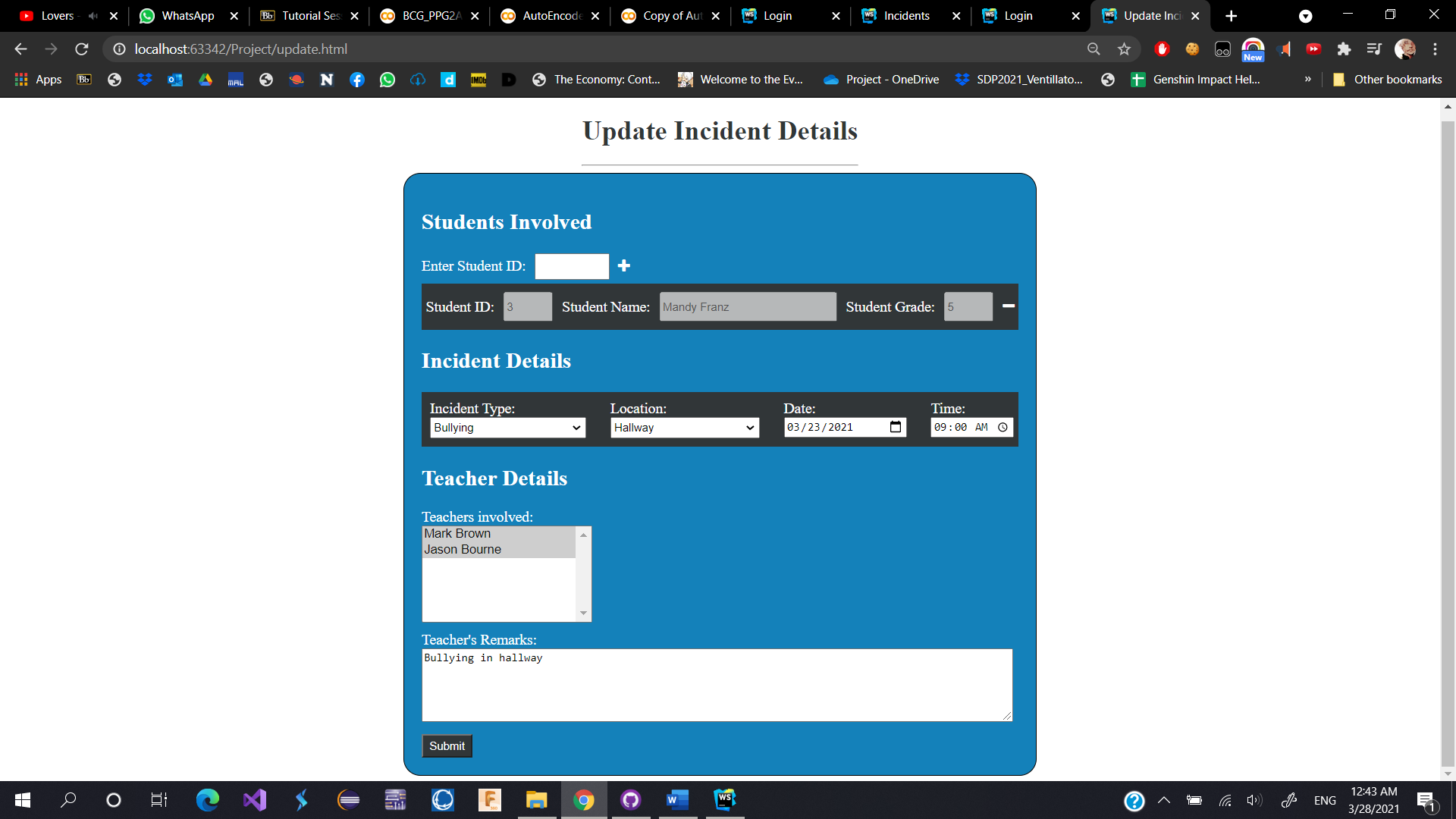
# 

# Report incident

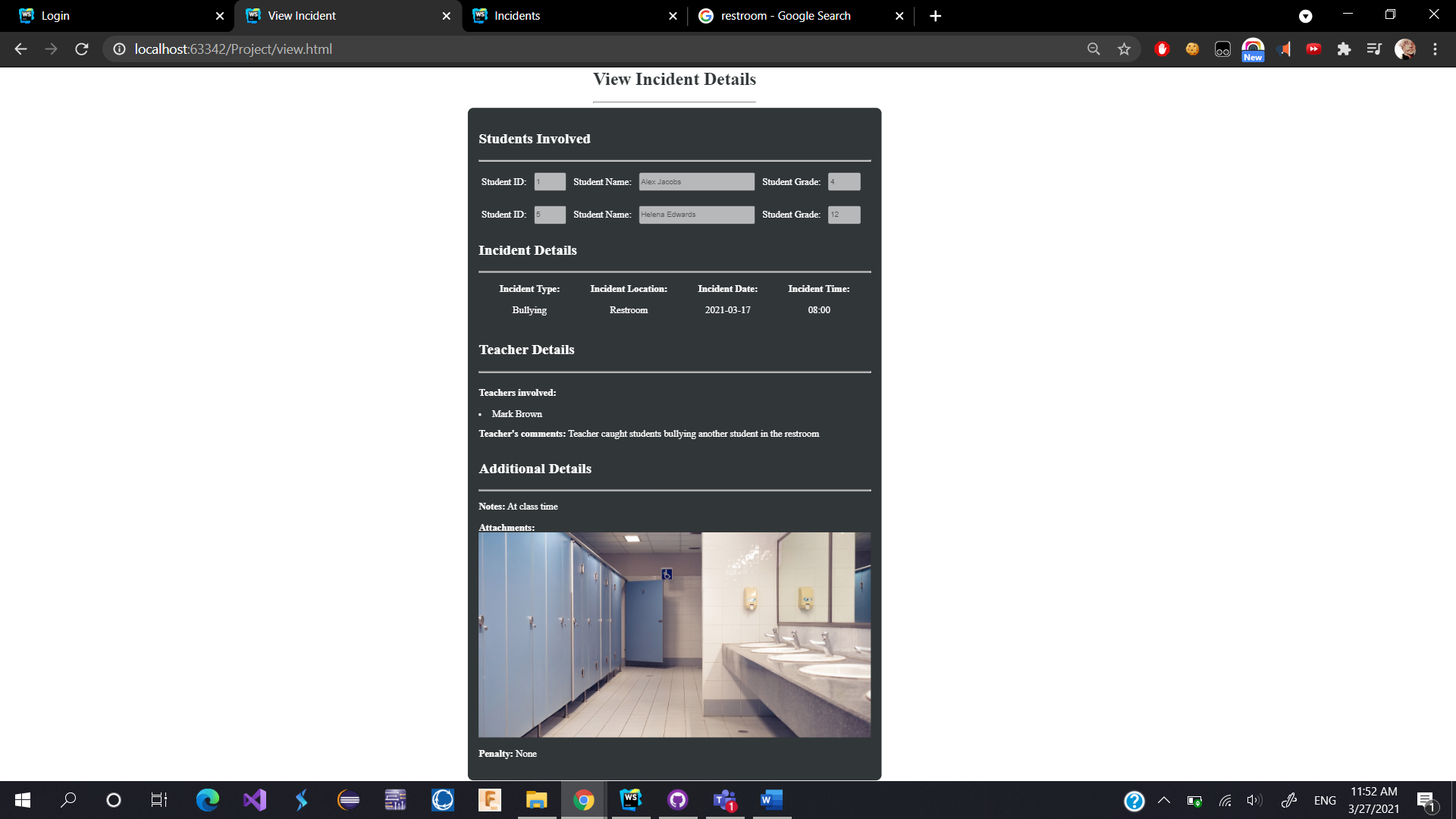




# Update Incident



# View Incident (including notes, attachments, and penalty)

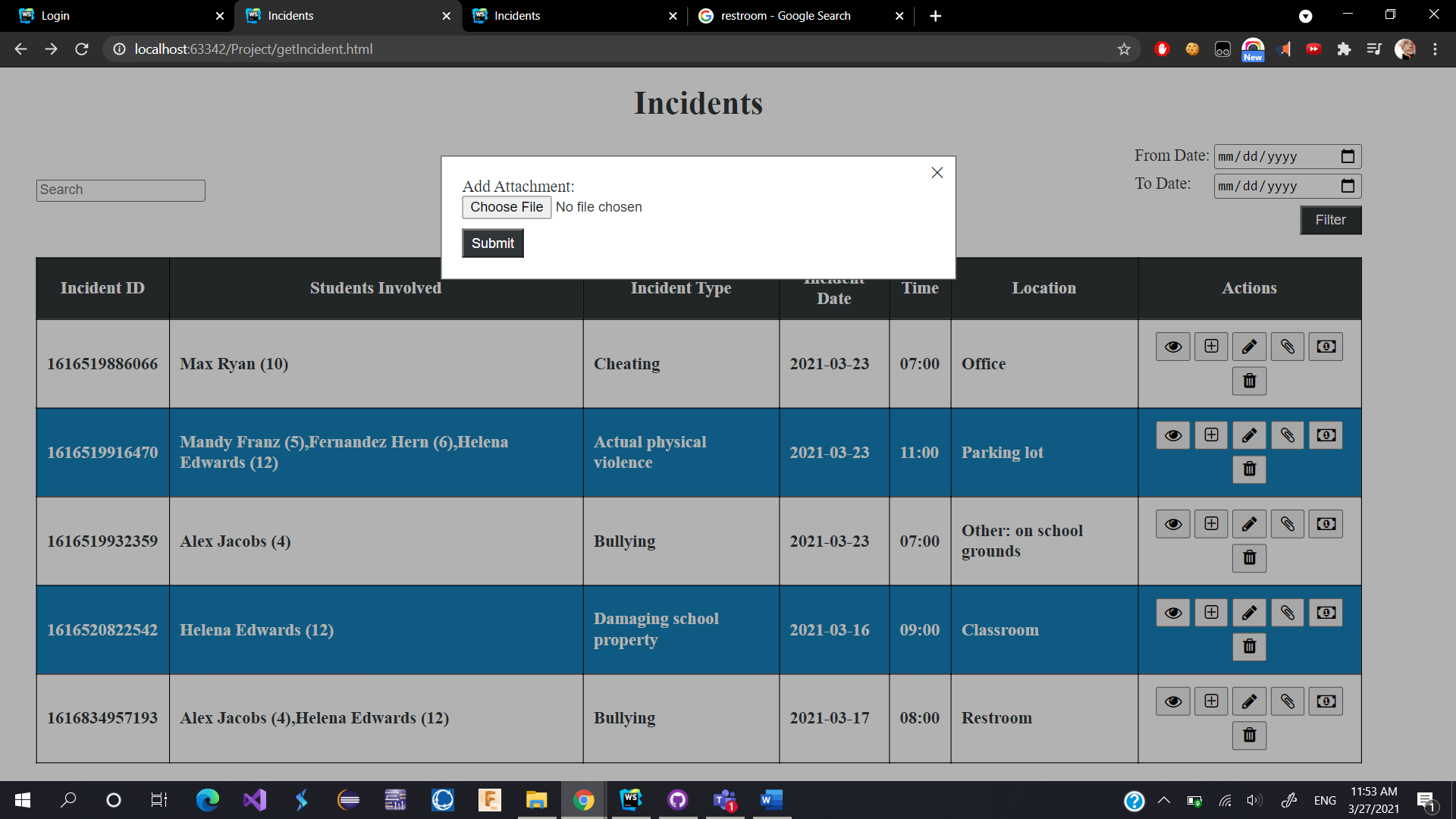


# Add Penalty

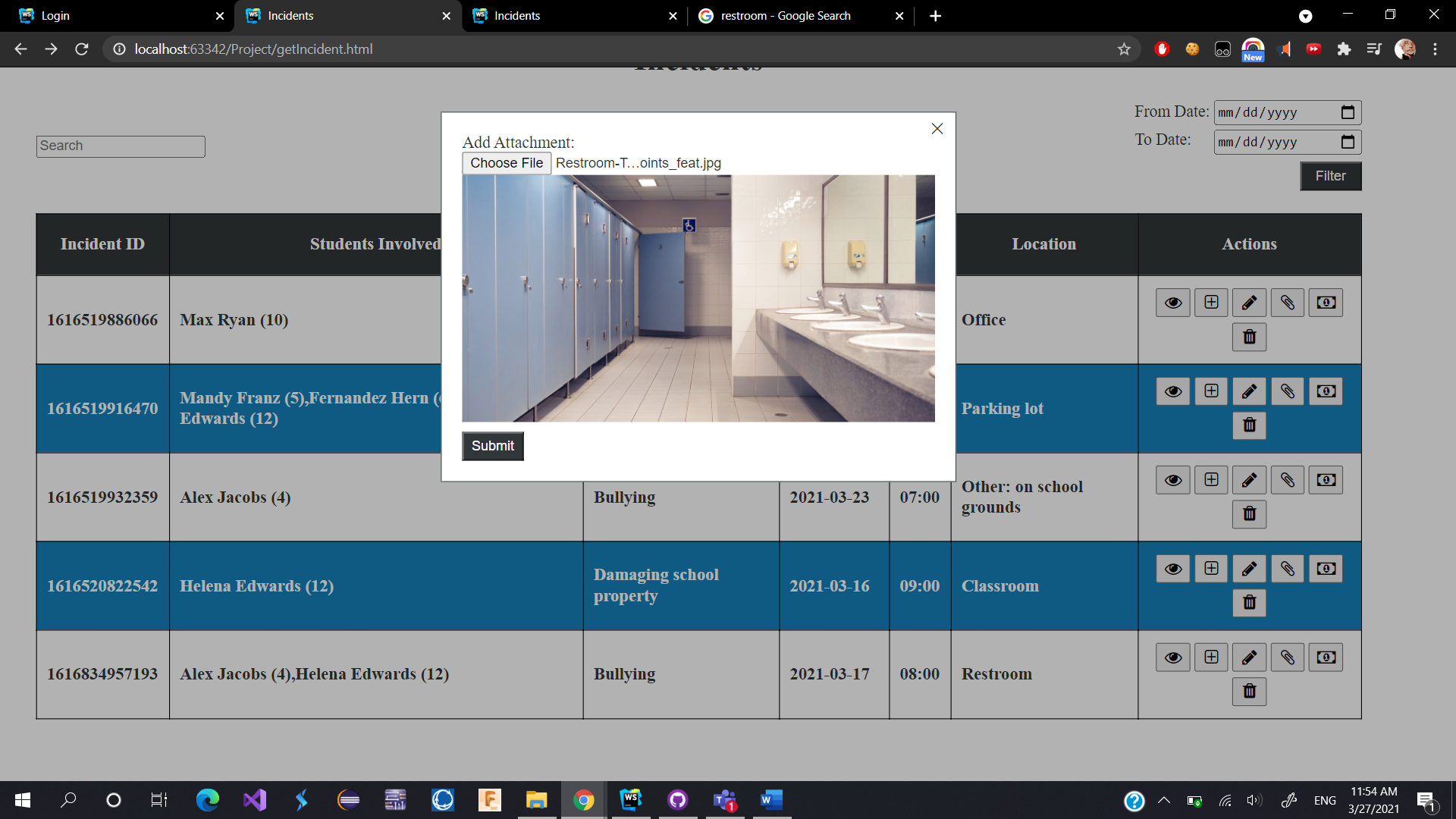


# Attachments (List, Add, Delete)

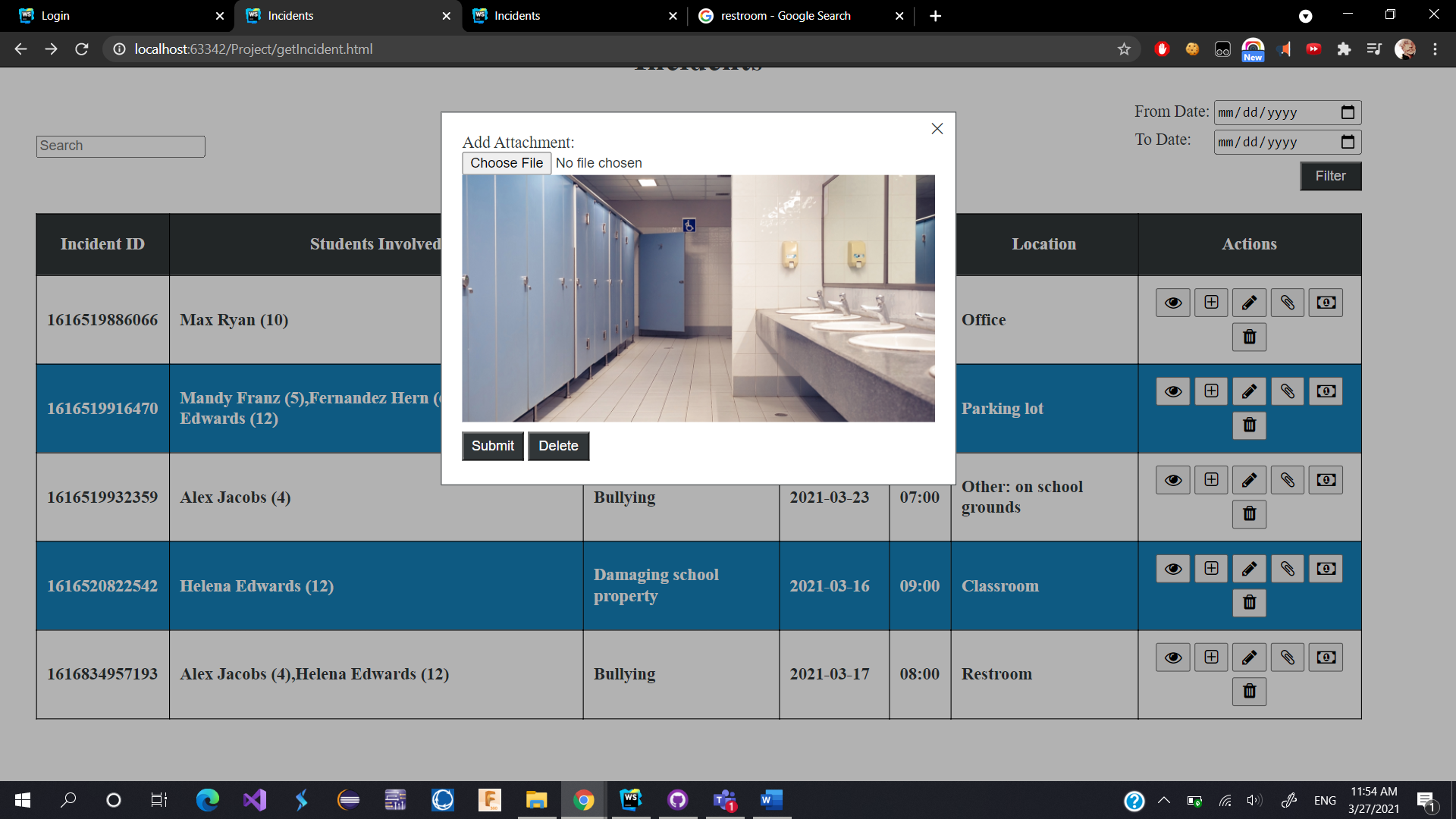
Click button to add attachment



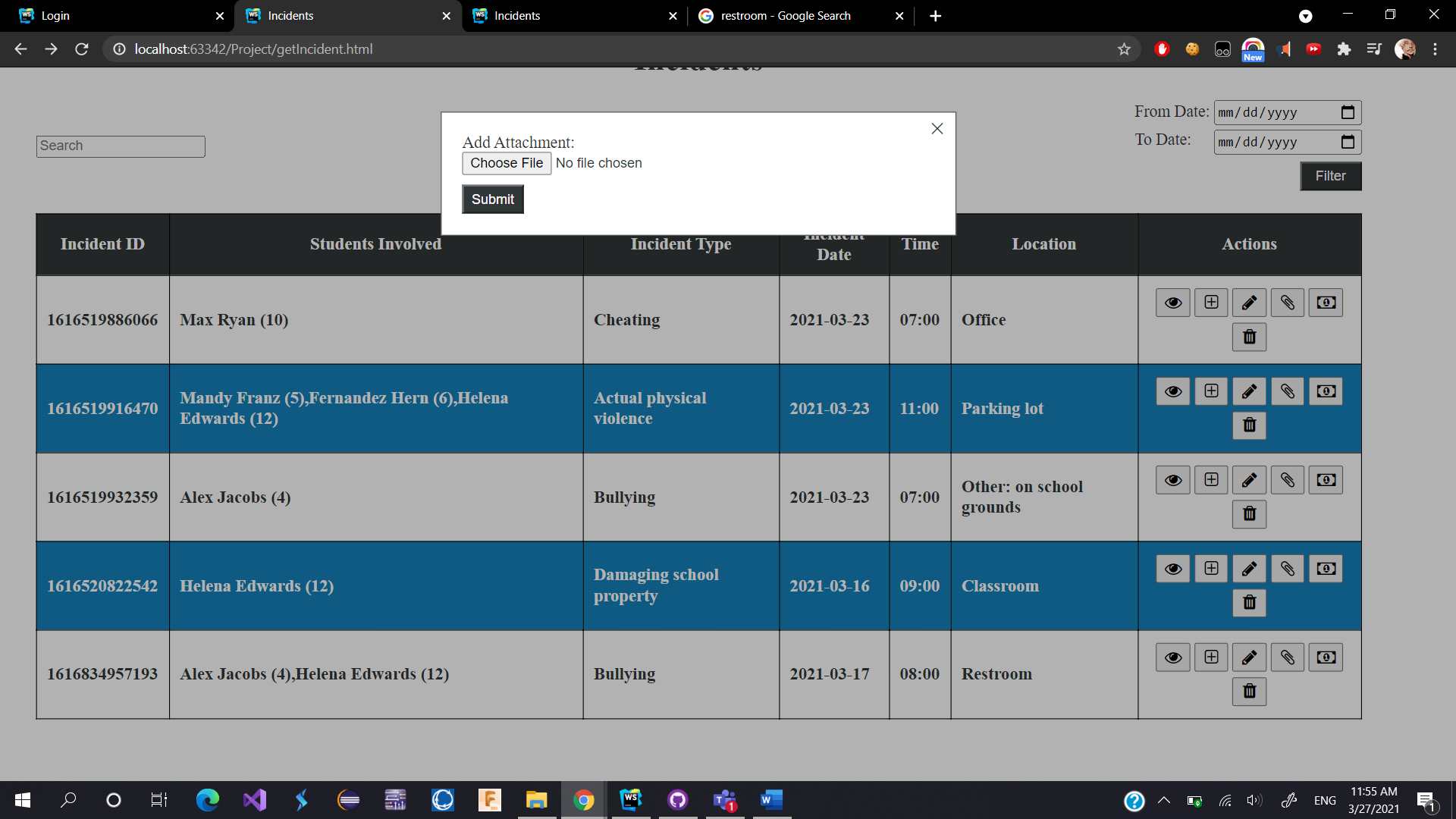
Choose image



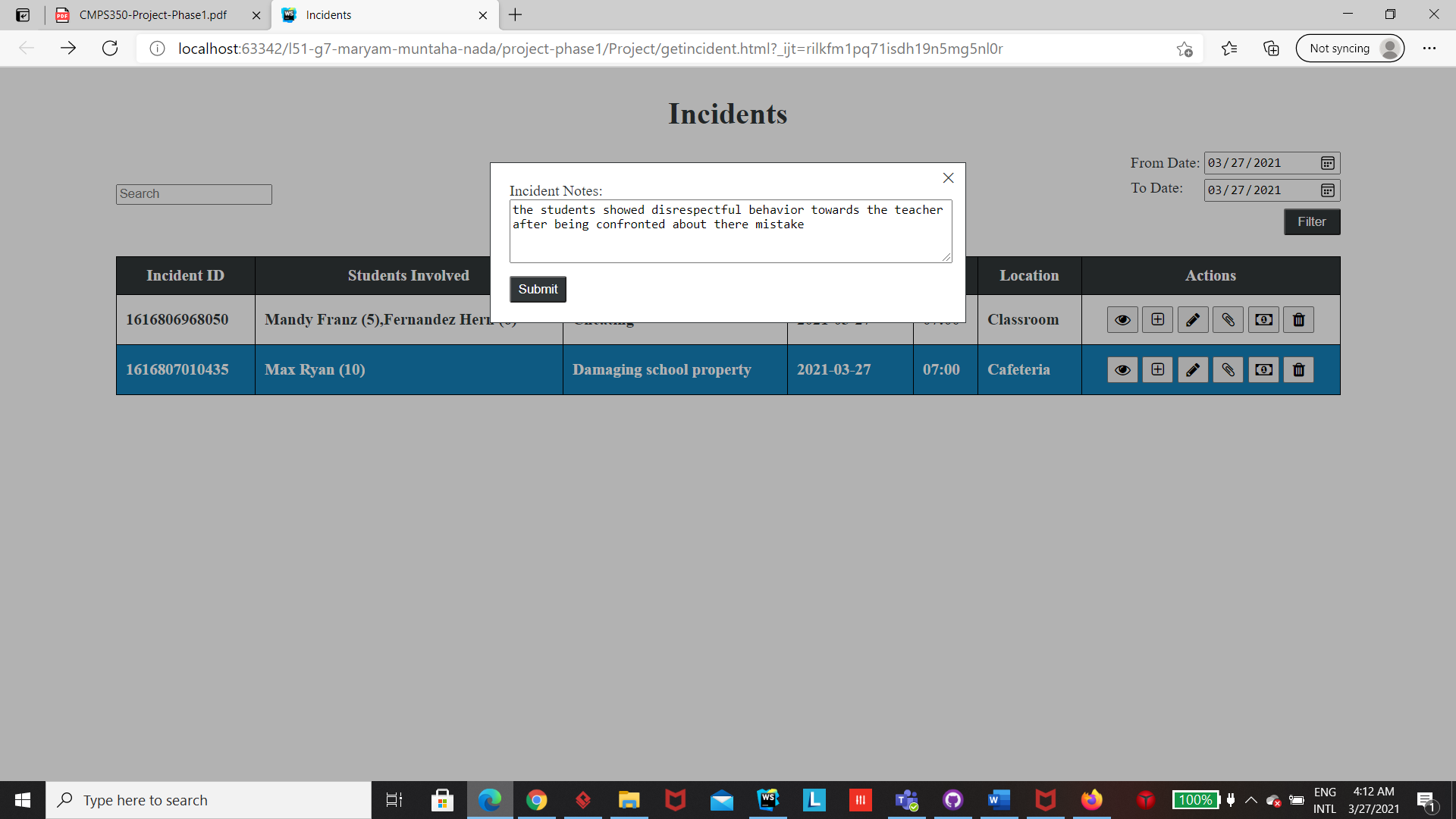
Submit image closes the modal, opening it again



Deleting the image



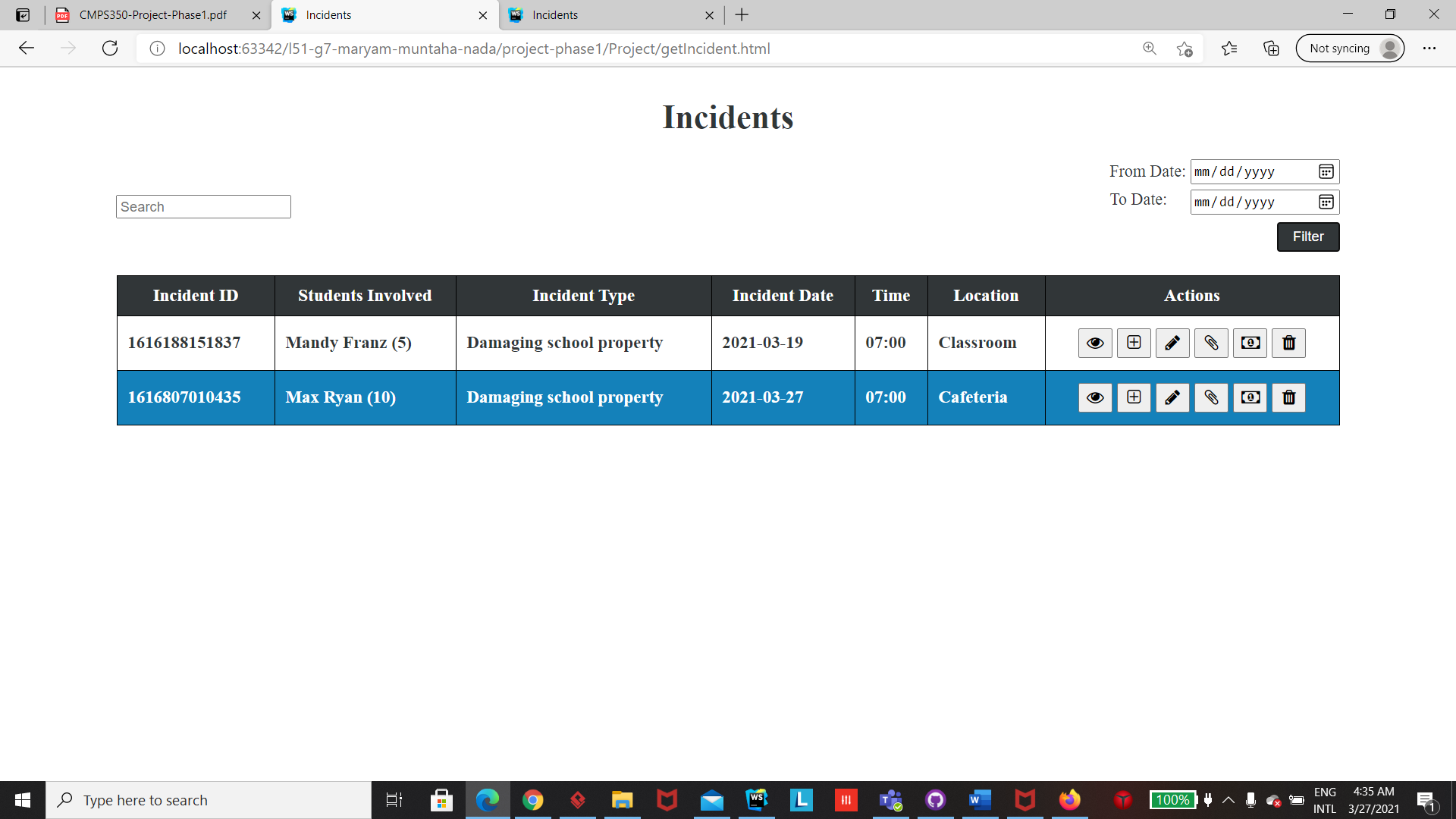
# Notes (List, Add, Delete)



# Incidents Summary Report

# 

# If you select any of the incidents you get the details



# Summary of team member contributions

Nada:

* Login
* Repositories
* JSON files and objects
* Report incident
* Update incident
* Get incident for parents
* View incident
* Add attachment
* Add penalty
* Incident summary report
* Modals
* All CSS

Muntaha:

* Add Notes
* Get incident (including search and filter by date)
* Delete incident
* Entities and repositories diagram
* Get incident and incident summary report HTML
* Most HTML