|  |  |  |
| --- | --- | --- |
|  | | |
|  | **CMPS 350 Project Phase 2 – Conference Management System (ConfPlus)**  **(15% of the course grade)** | |
| **Group Id:** | | G? |
| **Group Members:** | | Ahmed Ahmed (201907019)  Mohammed Al-Qeraisi (201909907)  Abdelbari Kecita (StudentId)  Mustafa Hussein (201906257)  **Emails:** student1@student.qu.edu.qa; student2@student.qu.edu.qa; student3@student.qu.edu.qa; |

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Criteria** | **Weight** | **Functionality**\* | **Quality of the implementation** | **Grade** |
| **Improvement over the first phase:**  *Depending on the implantation status of the previous phase, the following might apply:*   * *Completing missing functionalities;* * *Improving the design and implementation of paper submission: paper status, etc.* * *Improving the design and implementation of paper review: distinction between reviewed papers and papers to review, etc.* * *Various filtering possibilities for the conf schedule* * *Correct interpretation of session* * *Clarity of the various UIs.*   *By default, if no improvement is made, the student will have the same grade of previous phase for this category.*  *Everything you improve will add up to your previous grade that will be used as a baseline.* | 25 |  |  |  |
| Design and implement the Data Model.  Clarity of data entities, their attributes and relations (in Prisma and the conceptual model (the diagram)) | 10 |  |  |  |
| Init DB: populate the database with the data from the json files. | 5 |  |  |  |
| Repository Implementation to read/write data from the database | 10 |  |  |  |
| Database:   * The design and implementation of the statistics page * All other use-cases use the database, not JSON files or local storage. * All queries function correctly. | 40 |  |  |  |
| **Design and Testing Documentation**  **\* Design documentation:**  - 3 key lessons learned from Phase 1.  - Data Model diagram.  - UI Design table  - Data caching table  **\* Testing documentation:** with evidence of working implementation using snapshots illustrating the results of your solution testing (you must use the provided template).  \* **Discussion of the project contribution** of each team member [-10pts if not done] | 10 |  |  |  |
| **Total** | 100 |  |  |  |
| Bonus - successful deployment of the app and the Database to a cloud hosting service such as <https://vercel.com/> - successful implementation of use authentication. | 5 |  |  |  |
| Bonus- authentication through another service provider e.g. Google, Github, etc. | 5 |  |  |  |
| Copying and/or plagiarism or not being able to explain or answer questions about the implementation. | 0 |  |  |  |

# Application Design

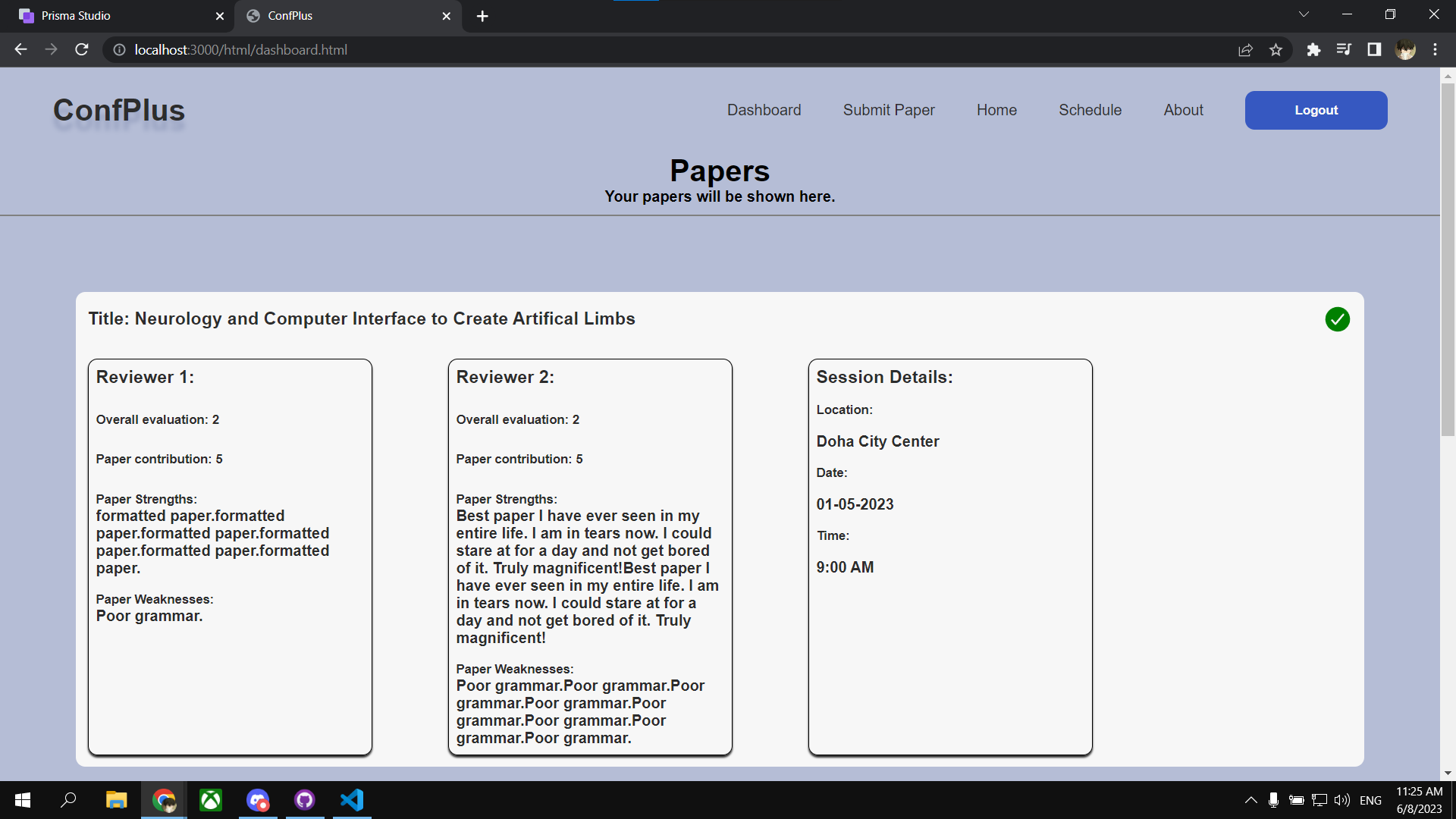
# Improvement over the first phase

# We faced alot of issues back in phase 1 with the UX (User Experience). The user can sometimes gets clueless about what to do, and what is the next step. So, we improved a lot on that side.

# 1- Dashboard page:

# The author gets an extra page called "Dashboard" to keep track of the status of his papers that he submitted.

# In the dashboard the author can see the status of the papers they submitted, each status has a special icon, like a checkmark when the paper is accepted. A card is clickable to see the details about the review itself and the location (When the paper gets accepted). A screenshot of a computer Description automatically generated



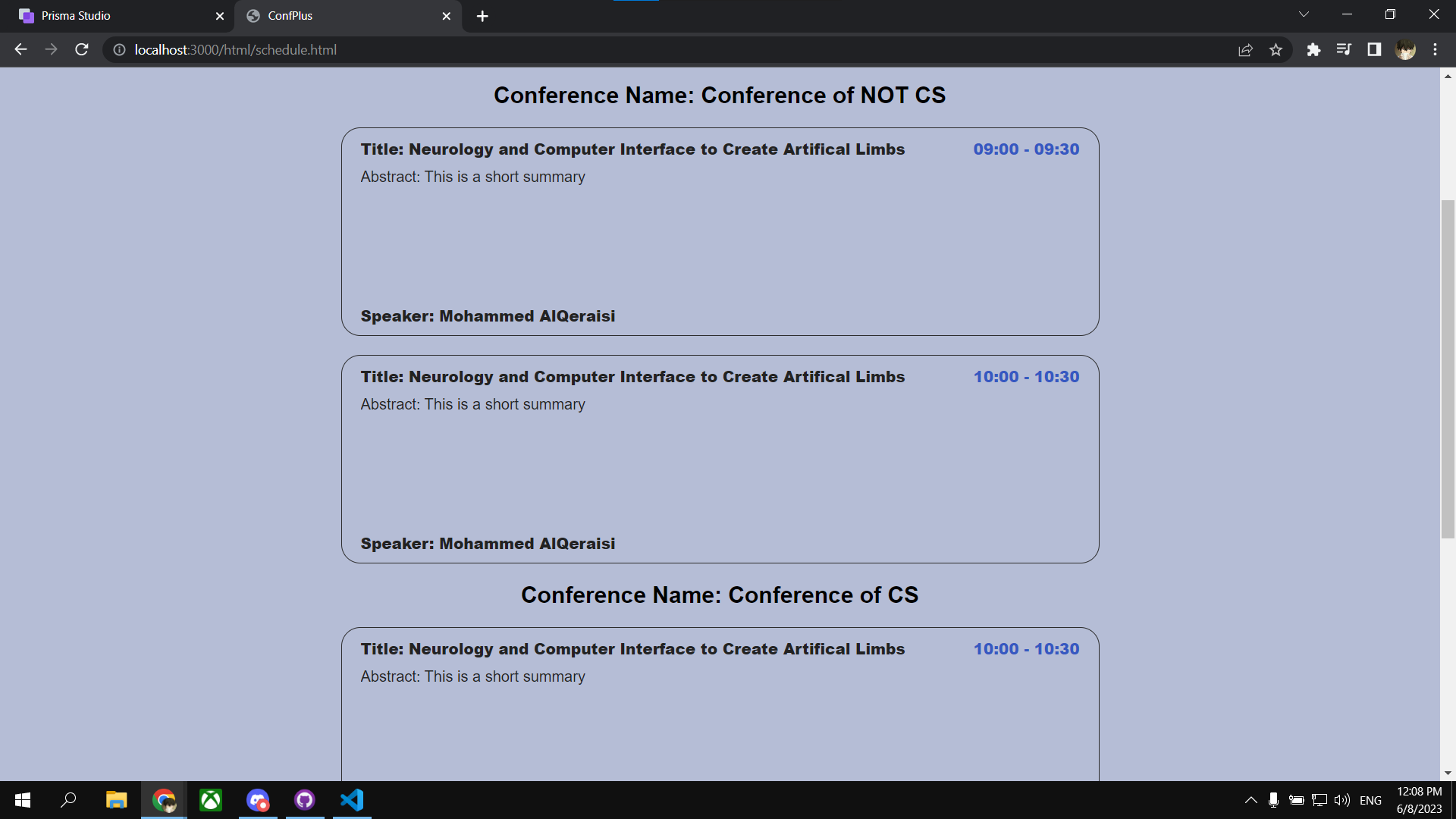
# 2- Review Page

# In the review page, the reviewer can distinguish between the papers that has been reviewed by them or not. If the paper is already reviewed the background of the card changes to the color blue, and the icon on the right gets filled. If it is not reviewed the background of the card becomes gray, and the icon gets outlined. We also added pagination at the bottom. This is an improvement for the UX.

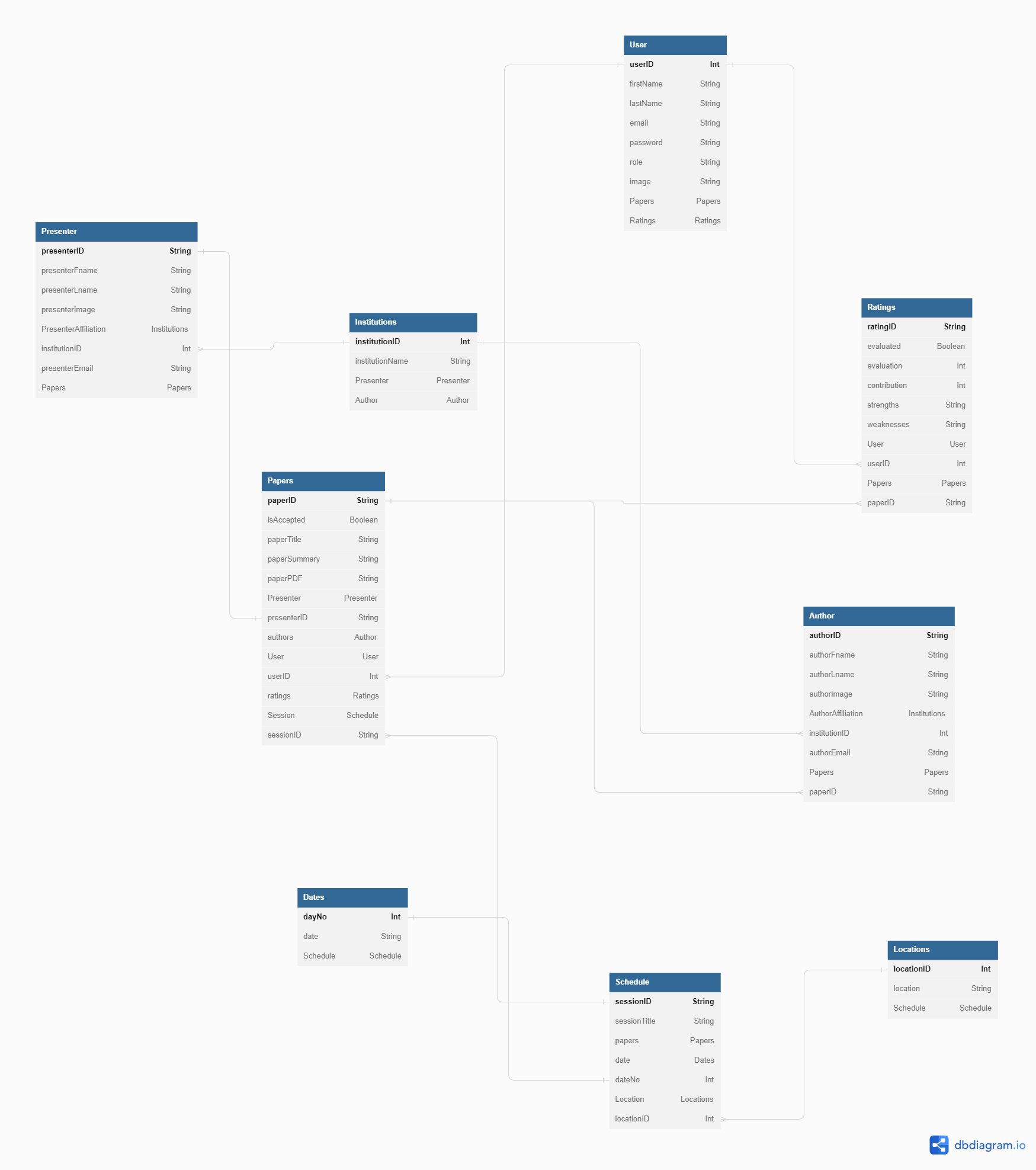
# 

# 3- Schedule Page

# In the schedule page, the user can see that the card is a paper in a certain session, and the card is labeled correctly to show the title and the abstract of that paper. If their is no filter applied, each set of papers in a certain session, will have the name of that session above those set of papers. Like in this example we see 2 papers in the first session, and 1 paper in the second session.



# Data Model diagram



# Database population

Give details and proofs with screenshots how you have populated database, the current content ..

# Database

Give details and proofs about how you have updated your application to use database. This applies to all use-cases.

Detail your statistics page

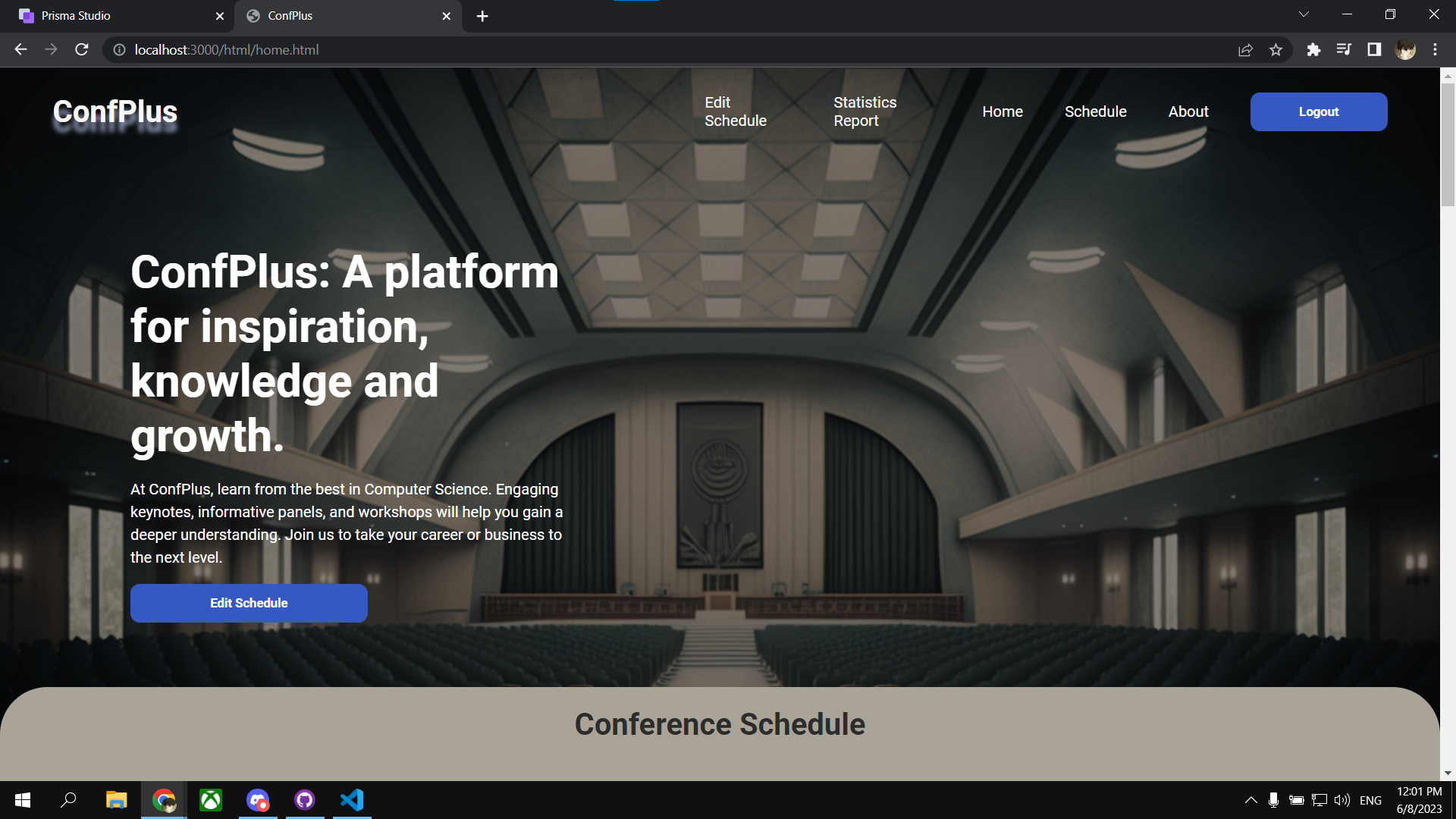
Give the list of your Prisma queries

# 3 technical lessons learned from your submitted solution vs. the model solution

# Testing

# Home

# In the home page we have divided it into section, The first section is the hero section which has a call-toa-action button that changes based on the user that is logged in, In this example the organizer is logged in, so he is shown "Edit Schedule". The second section is the main that contains some info about the schedule, and the staff behind this conference, and an accordion at the bottom to show FAQs about the conference. The last section is a footer.



A screenshot of a computer

Description automatically generated with low confidence

A screenshot of a computer

Description automatically generated

# Custom Login

# 

# Submit paper

# In the submit paper page, the author can submit his paper details, also they can fill in their info and the other authors' info. The first info to get filled will be the presenter info. There is a button to add extra authors or delete them.

# 

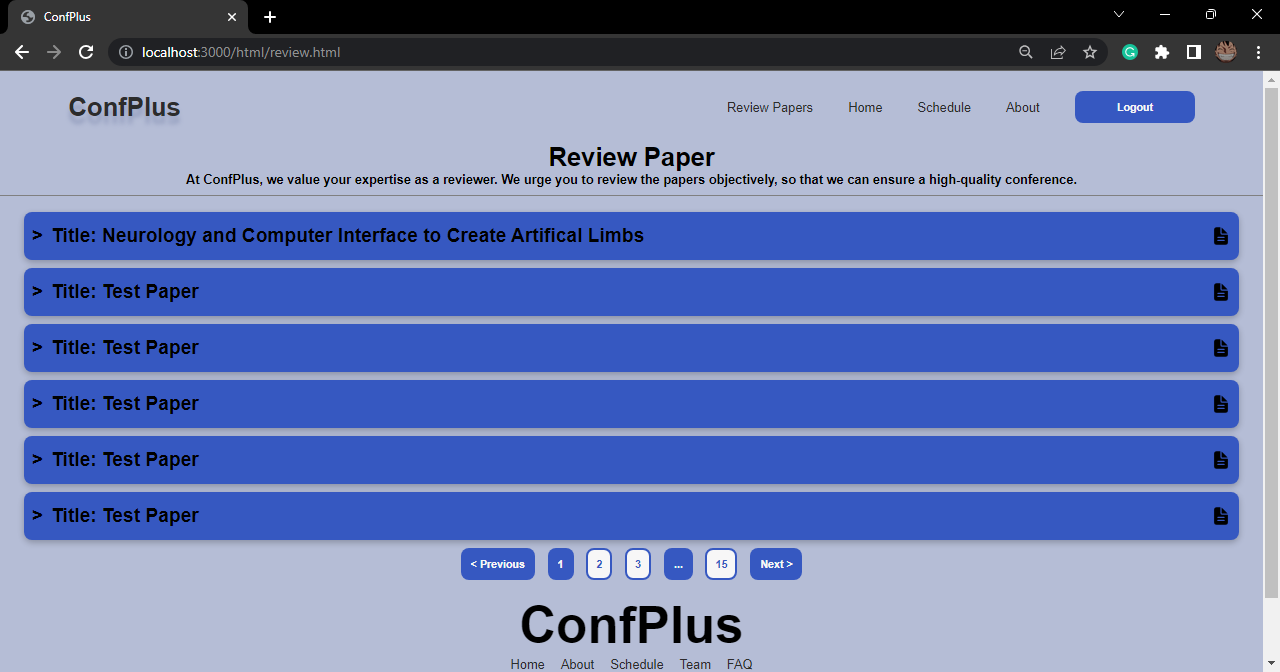
A screenshot of a computer

Description automatically generated with medium confidence  
A screenshot of a computer

Description automatically generated with medium confidence

# Review paper

# In the review page, the reviewer is first shown a list of the papers that are assigned to them, if the paper is not reviewed yet, the background will be gray, otherwise it will be blue.



# When clicked on a paper to be reviewed, the reviewer will be shown a picture of the presenter and their info, and the details of the paper, like the title, the abstract and the authors list. Then a form at the bottom to evaluate the paper.

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated

# Get conference schedule

# When the schedule page is first loaded all the papers will be displayed regardless of their date, but each set of papers will have the name of their conference above them. Also the user can filter the schedule based on the date. When the filter is applied, the name of the conference will disappear.

# A picture containing text, screenshot, software, computer icon Description automatically generated

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

# Edit conference schedule

# In the edit schedule page, the organizer is shown an "add session" button when the schedule is fully empty. But when there are sessions available, the organizer can simply choose from them or add a new one.

# When the organizer clicks on one session to edit it, he is shown a form that gets filled with the session's info, and 2 buttons to add a paper to this session, or delete this session entirely.

# When adding a paper to the session, the organizer can select the time of its presentation on that session. A screenshot of a computer Description automatically generated

# After adding all the papers and select their time of presentation, the organizer must click "Update" to update the info of the session.

# 

# Conference Statistics Report

# In the statistics page, we used simple math operations to perform the calculations of different statistics in this page.

# 

# Discussion of the project contribution of each team member

|  |  |
| --- | --- |
| 1. Team Member | Contribution Percentage |
| Ahmed Ahemd | 25% |
| Abdelbari Kecita | 25% |
| Mustafa Hussein | 25% |
| Mohammed Al-Qeraisi | 25% |

# Team coordination:

# To collaborate efficiently, we used Discord as our primary communication tool, which allowed us to discuss project-related issues and share our screens to resolve problems in real-time. We also utilized GitHub to manage our codebase, which enabled us to work on different parts of the project simultaneously and merge our changes seamlessly.