CMPS 350 Project Phase 1 – WebApp UI Design and Implementation
Conference Management System (ConfPlus)
(20% of the course grade)

Group Members:	Osama Hardan (202107304)
	Ibrahim Alsalimy (202104709)
	
	Emails: oh2107304@student.qu.edu.qa;
	ia2104709@student.qu.edu.qa;

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	Your Grade
Application Design: Entities, Repositories and Web API class diagrams, flow diagrams	10%			
Complete and correct implementation of the requirements:	**80%**			
• Login	10			
• Submit paper	20			
• Review paper	20			
• Create/update conference schedule	20			
• Get conference schedule	10			
Testing, documentation, and group work: - Use screen shots to illustrate your tests. - For every team member detail the list of accomplished tasks by the member, **and** the overall contribution percentage to the project (%). - Team coordination: describe in a concise way how the team members collaborated to achieve the project. List the collaboration tools that you have used, if any **All of these elements should be reported in the template below**	10%			
Total	100			
Copying and/or plagiarism or not being able to explain or answer questions about the implementation	-100%			

* **Possible grading for functionality** - ***Working*** (get 70% 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.

1. Current status of the project implementation

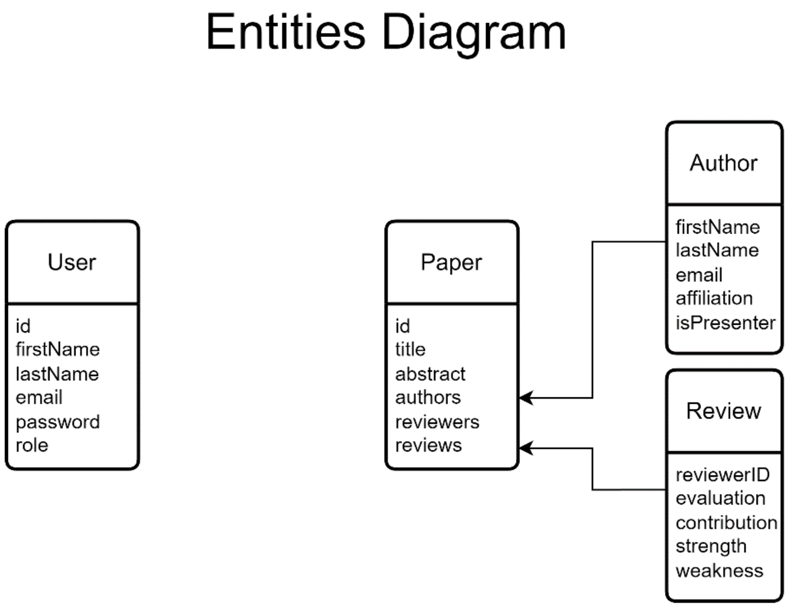
1- Completed Use Cases:
Use Case 1: Login.
Use Case 2: Submit a Paper.
Use Case 3: Review a Paper.

2- Partially implemented use-cases / functionalities:
Reading locations from the json file (from use case 4).

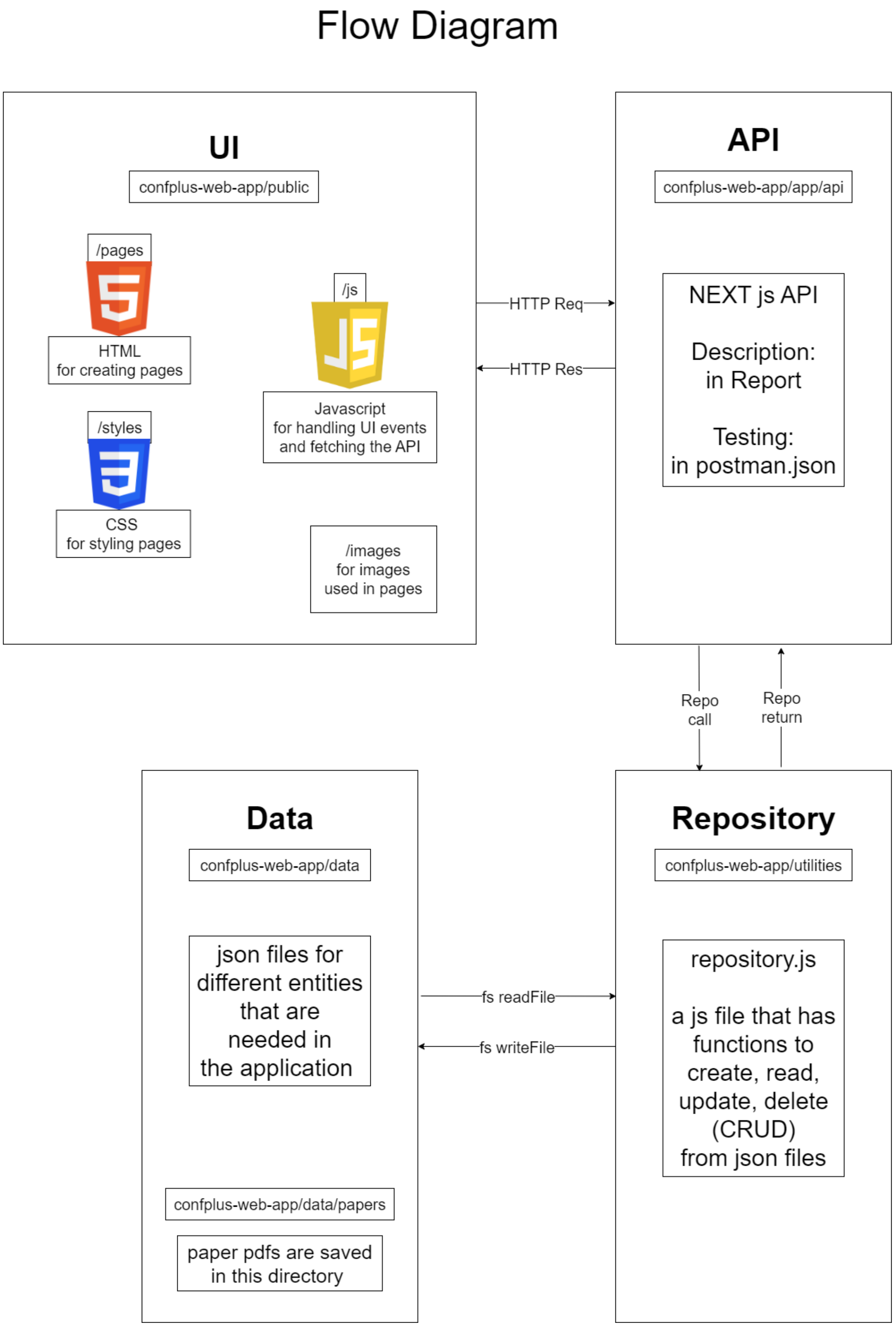
3- Use Cases that have not been implemented:
Use Case 4 and 5

2. Application Design

2.1. Entities class diagram



2.2. Flow diagram



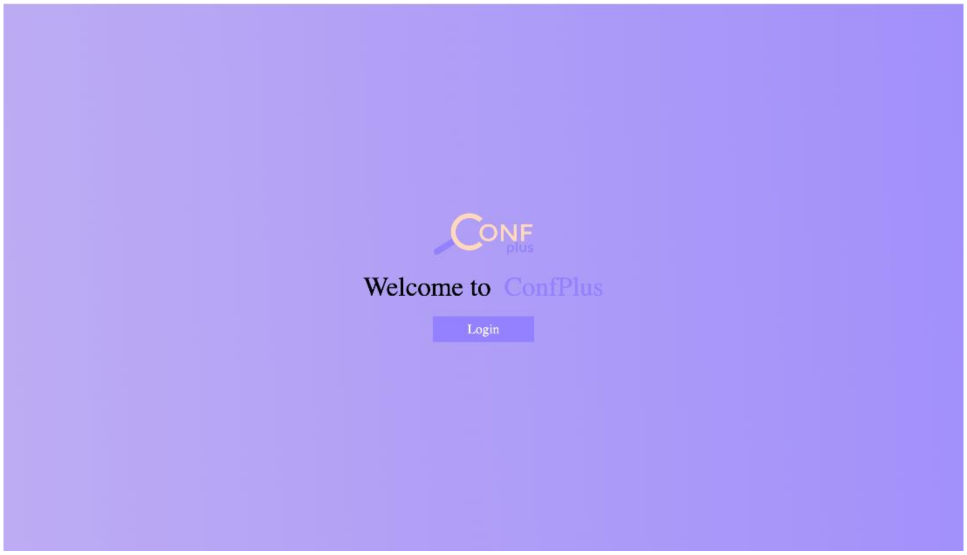
2.3. Web API Description

Method	URL	Description (response status)
GET	/api/users?email=e&password=p	Returns the user object if the credentials are correct (200) by searching for the email in users.json and rejects unfound email (404) or wrong password (401).
GET	/api/users/:role	Returns an array of the users whose role is role (200), if not found it returns a message (404).
GET	/api/papers	Returns an array of all the papers in papers.json (200). if no papers it returns a message (404).
GET	/api/papers?reviewerID=rid	Returns an array of all the papers in papers.json that are assigned to a reviewer using his id. if no papers assigned it returns a message (404).
POST	/api/papers	Creates a new paper and saves it to papers.json and saves the pdf to data/papers (201). The paper details (title, abstract, authors, and pdf file) are sent in the request body using FormData. It also assigns 2 random reviewers from users.json.
GET	/api/papers/:id	Returns a paper object using id (200). Returns a message if not found (404).
GET	/api/papers/:id/pdf	Returns the pdf file of the paper that has id from data/papers. as a Blob object (200). And returns a message if no pdf for that paper (404).
GET	/api/papers/:id/reviews?reviewerID=rid	Returns the review of the reviewer who has reviewerID in the paper that has id from papers.json (200). And returns a message if no review found (404).
POST	/api/papers/:id/reviews	Creates a new review or updates it if it exists and saves it in the paper json object in papers.json (201 if create, 200 if update). The review details (reviewerID, evaluation, etc…) are sent in the request body.

GET	/api/institutions	Returns an array of all the institutions in `institutions.json` (200).
GET	/api/locations	Returns an array of all the locations in `loactions.json` (200).

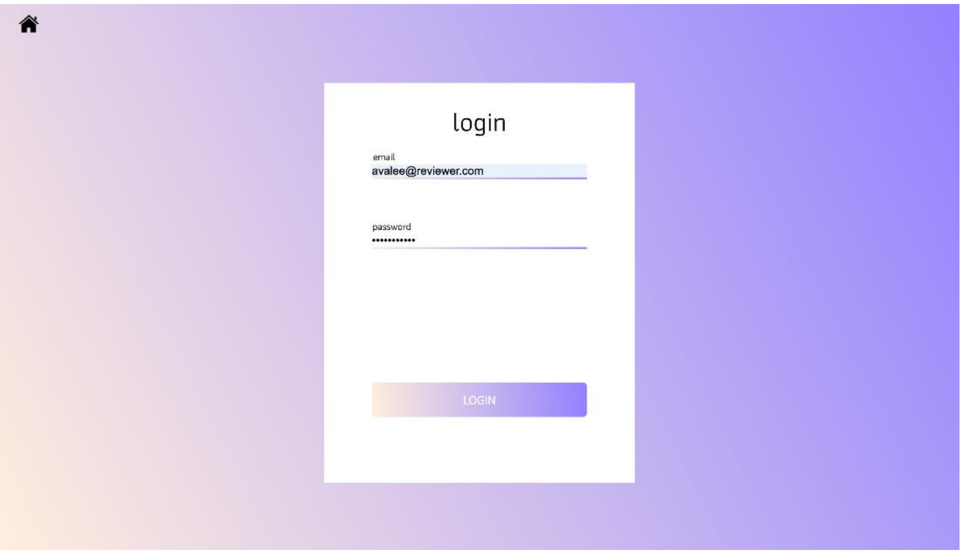
3. Testing

3.1. Login



Welcome to ConfPlus

Figure 1 Home Page



email
avalee@reviewer.com

password
•••••••••••

Figure 2 Login Page

3.2. Submit paper

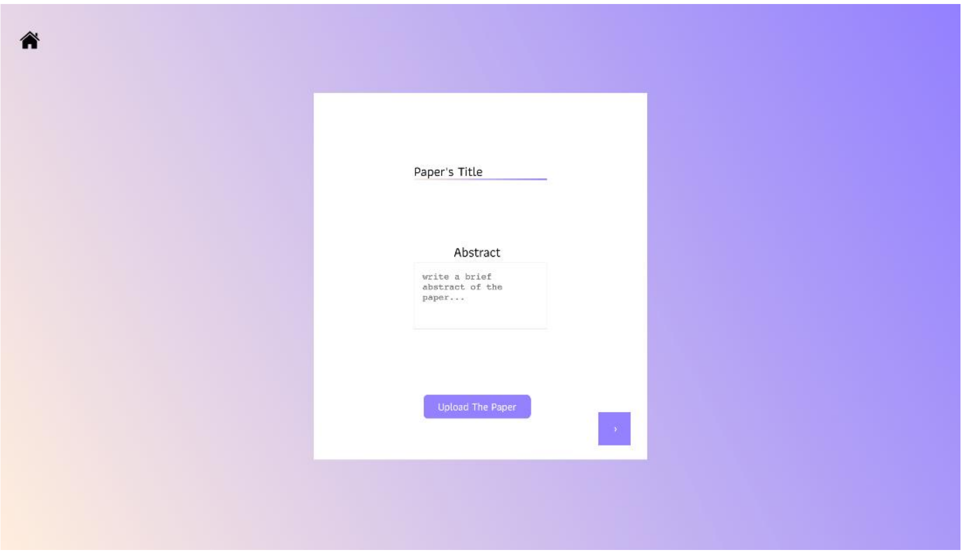
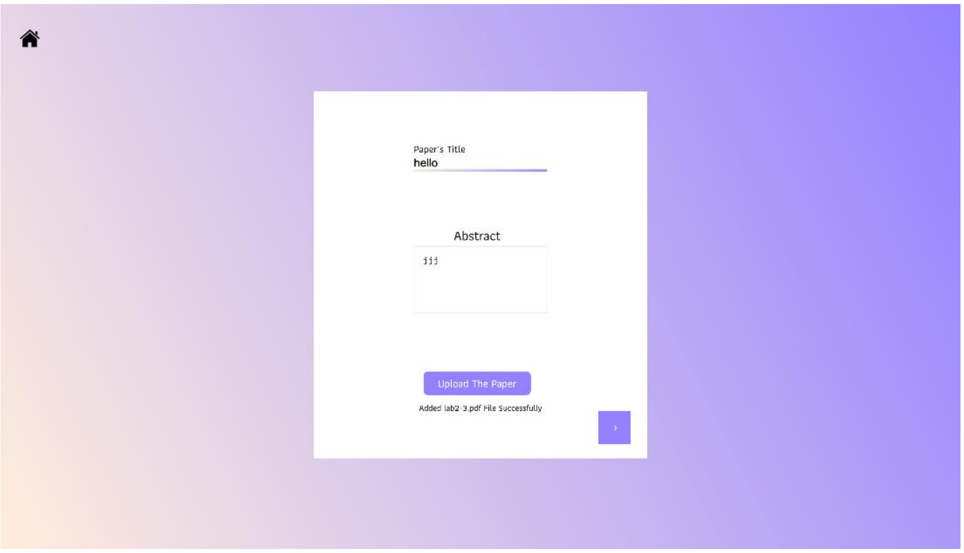


Figure 3 Submit Paper Stage 1 initial state

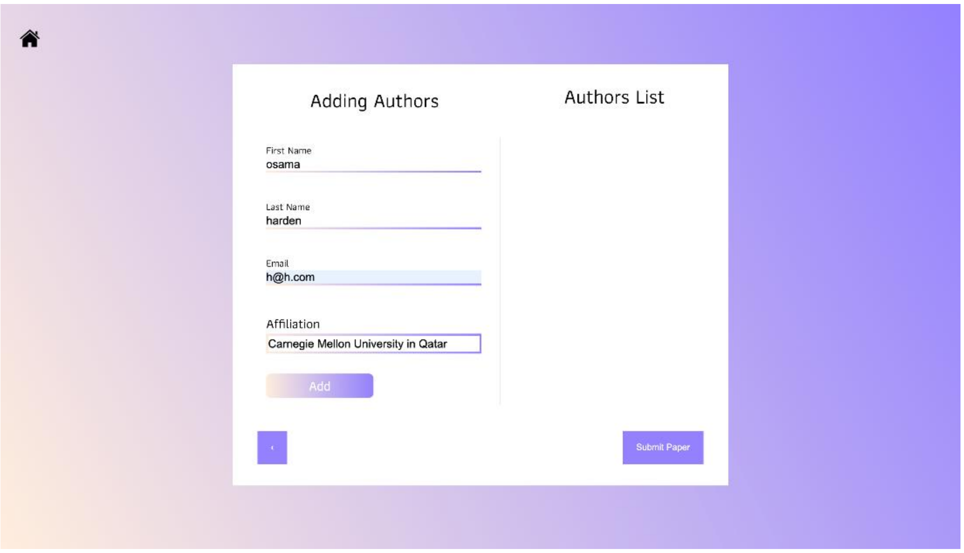


Paper's Title
hello

jjj

Added lab2-3.pdf File Successfully

Figure 4 Submit Paper Stage 1 after filling inputs



Adding Authors

Authors List

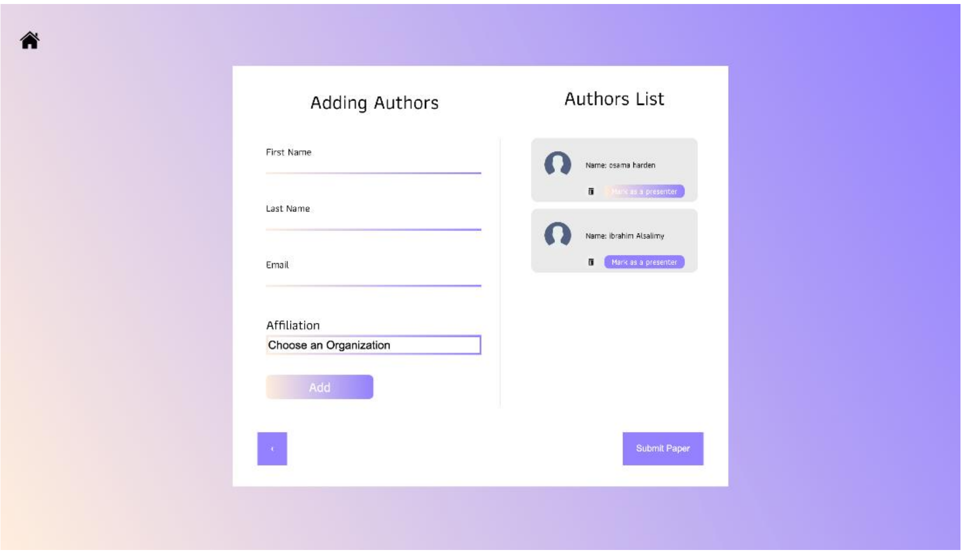
First Name
osama

Last Name
harden

Email
h@h.com

Affiliation
Carnegie Mellon University in Qatar

Figure 5 Submit a Paper Stage 2 initial state



Adding Authors

Authors List

First Name

Last Name

Email

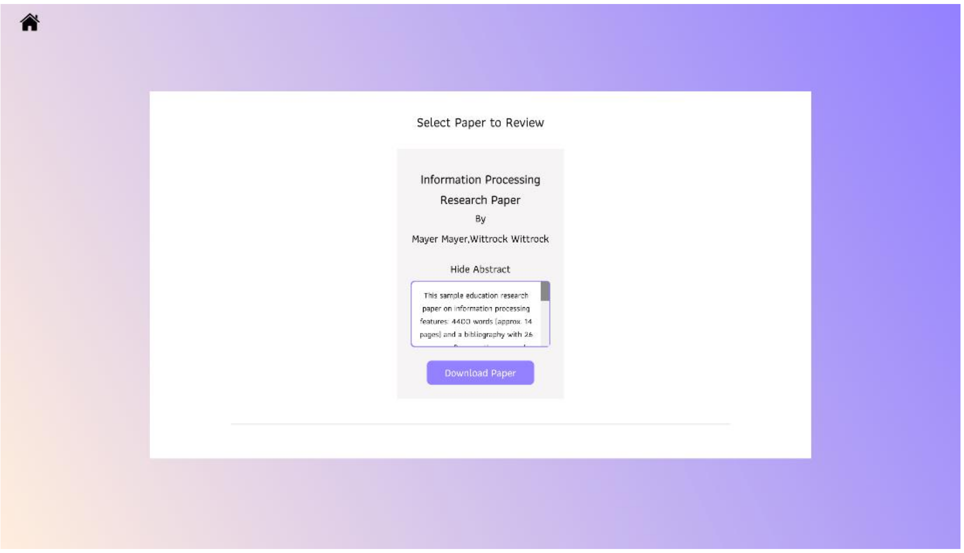
Affiliation
Choose an Organization

Name: osama harden

Name: ibrahim Alsalimy

Figure 6 Submit a Paper Stage 2 after filling inputs

3.3. Review paper



Select Paper to Review

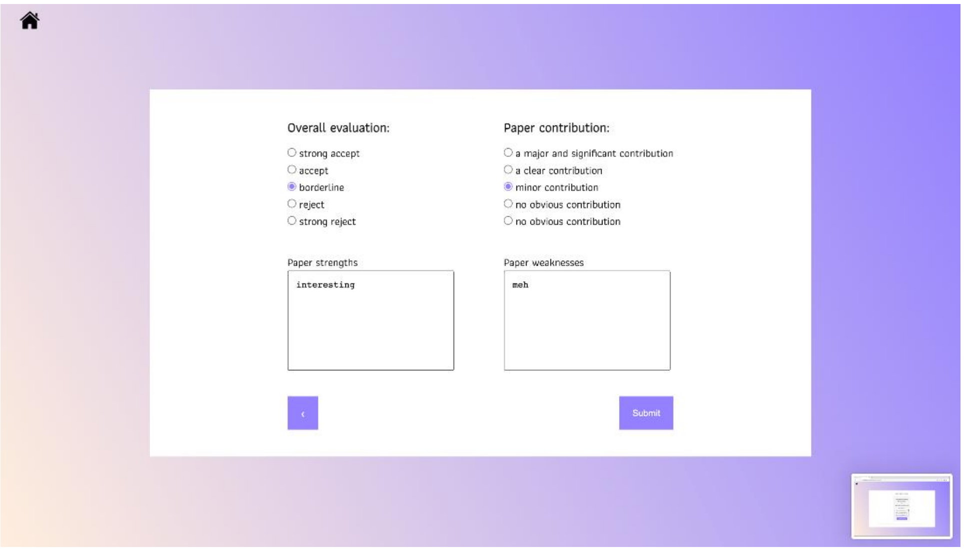
Information Processing Research Paper

By

Mayer Mayer,Wittrock Wittrock

This sample education research paper on information processing features: 4400 words (approx. 14 pages) and a bibliography with 26

Figure 7 Reviewer chooses a paper to review



Overall evaluation:

○ strong accept
○ accept
● borderline
○ reject
○ strong reject

Paper contribution:

○ a major and significant contribution
○ a clear contribution
● minor contribution
○ no obvious contribution
○ no obvious contribution

Paper strengths

interesting

Paper weaknesses

meh

Figure 8 Review Paper Form

4. Discussion of the project contribution of each team member

4.1. Team member contributions

1- Osama (50%): I worked on the API part designing and implementing the API routes. Also, I did the repository and decided how the files are accessed. I also worked in client JavaScript files in the parts that need fetching the API since I am the one who implemented it.

2- Ibrahim (50%): I made the designs in Figma after learning the fundamentals of UI/UX design, I also made the HTML and the responsive CSS for all pages and with extra cool transitions and animations I saw on the internet and tried to implement it myself. Finally, I made some JS events like onclick, onfocus, onblur events.

4.2. Team coordination

- Used GitHub repo to develop the project.
- Used Trello to write the tasks and track its progress.
- Used Figma to design the UI and the Logo.
- Used a discord server to communicate and conduct meetings.

We achieved this progress by strictly defining each one area of work. Ibrahim for UI and Osama for API and files. This strategy helped us not to have a lot of conflicts. However, that does not mean each one worked alone, we were taking each other’s opinion into consideration and in the regular meetings we tried to update each other with the progress in his area.