**FACULTY PUBLICATION REPOSITORY**

**GROUP 12**

|  |  |  |
| --- | --- | --- |
| **Members** | **Roll Number** | **Role** |
| **Sharan Babu** | **CB.EN.U4CSE19207** | **Tester** |
| **D Vishal Reddy** | **CB.EN.U4CSE19213** | **Business Analyst** |
| **R S Vishal** | **CB.EN.U4CSE19237** | **Software Engineer** |
| **T Selva Sanjana** | **CB.EN.U4CSE19250** | **Software Engineer** |
| **V Srijay** | **CB.EN.U4CSE19256** | **Project Manager** |

**ABOUT THE PROJECT**

Over the past few decades, the number of publications being published by university faculty all over the world has exponentially increased, mainly due to the digitization of publishing. This makes finding relevant ones a very difficult task, leading to problems such as research gaps or redundancies, where multiple publications are published on the same topic, due to the authors of one not knowing about the existence of the other. To solve such problems, we need a common repository that stores this massive number of publications and also provides additional functionalities such as grouping by topics, or tracking all the publications of a single author. This is what our project aims to do, a single repository for all faculty publications, where inconsistency and unavailability of information will become a thing of the past.

**STAKEHOLDERS**

* Faculty
* Students
* Research Scholars

**MODULES**

* Authorization
* Add/Delete a publication
* Search
* User Permissions (View/Edit)
* Admin

**ASSUMPTIONS**

* The project will be implemented as a web-application which is supported in PC’s and mobiles.
* The website will not deal with any payments.
* The publications are already registered and verified.
* Publications cannot be edited in the repository.

**TECH STACK**

* MERN - MongoDB, Express, ReactJS, NodeJS

**PROJECT MANAGEMENT TOOLS**

* Jira
* Microsoft Teams

**TESTING TOOLS**

* Quality center

**REPORTS**

1. Requirement report - Contains the requirement of stakeholders.

2. Testing report - Includes modules with errors and modules that work correctly.

3. Debugging report - Errors that are rectified.

4. User feedback-Users likes and dislikes about the software.

**EPICS, USER STORIES AND TASKS**

**Epic: Research paper (form) manipulation**

**User story 1:** As user of the repository, I should be able to upload my research paper to the repository, so that it is available for everyone to benefit from it.

**Sub-Tasks:**

* Provide a button which redirects the user to the upload page
* Design a page to upload and include details relating to the research papers
* Implement upload page HTML/CSS/JS
* Ensure that all the details and files provided are safely stored and secured in the database
* Display a success message
* Redirect to the uploaded file
* If the user is not registered restrict the user from uploading items
* Display a list view of all the uploaded files

**User story 2:** As a user of the repository, I should be able to update/edit any details entered regarding my submissions, so that I can correct any mistakes if they were made in the process of uploading the research paper.

**Sub-Tasks:**

* Provide a button which directs the user to the update page
* Design a page to update/edit the details which were provided during uploading the files
* Upon the clicking of the edit button the user should be directed to the edit page which will have a form with the details of the item which you want to edit
* Implement update page HTML/CSS/JS
* Design a prompt to confirm the updates and display a successfully updated message after updating
* Ensure that all the details and files provided have been updated in the database
* Ensure that the changes have been reflected in the repository
* Redirect to the list view of your items

**User story 3:** As a user of the repository, I should be able to delete any entries that I have made at any time, so that I can prevent other users from accessing it.

**Sub-Tasks:**

* Provide a button which allows the user to delete their own paper
* Design and implement a prompt which asks for confirmation to delete the item
* Upon deletion, display a successfully deleted message
* Ensure that the deleted item has been removed from the repository
* Ensure that all the deleted items have been removed from the database
* Redirect to the list view which displays all the papers

**User story 4:** As a user of the repository, I should be able to view all of the research papers that are available on the repository, so that I can read the research papers.

**Sub-Tasks:**

* Design a main page which allows the users to view all the available papers
* Implement the page using HTML/CSS/JS
* Ensure that all the links and buttons are responsive
* Redirect to the file which the user wants to view upon its selection
* Design and implement a page which is opened upon selection of an item
* Users who are viewing items should not have access to update and delete buttons
* Users who are viewing their own items should have access to update and delete buttons

**User story 5:** As an administrator, I should be able to delete any files that are classified as spam, so that I can keep the repository clean and free from spam.

**Sub-Tasks:**

* Provide special administrative roles to specific users
* Provide these users with special functionalities
* Ensure that the files deleted by the admin should be removed from the database
* Ensure that the changes have been reflected in the repository
* When the admin deleted a paper, the owner of that paper should get a notification specifying the same
* Admins can also remove users if they are found to be misusing the repository
* These users can delete any of the available papers in the repository and can delete users, hence delete buttons should be made available on every item while in admin mode
* Design and implement prompts which ask for conformation before deletion

**Epic:** **Feedback/reviews on research paper**

**User story 1:** As a faculty, I want to give feedback on the research papers presented by other faculties

**Sub-Tasks:**

* **Design a tab below the research paper to view the reviews/feedbacks on the paper**
* **Create a button to add a comment**
* **Set user permissions to allow to edit/delete a comment**
* **Provide a button to reply to a comment**
* **Create SQL scripts to store the comments in the database**
* **Implement review page using HTML/CSS/JS**

**User story 2:** As a faculty, I want to add comments to my own research paper, for conveying information about how to read this paper

**Sub-Tasks:**

* **Create SQL scripts to add/edit/delete a comment**
* **Implement review page using HTML/CSS/JS**
* **Provide a button to add/edit a comment**
* **Research and decide on unit test and mocking framework**
* **Design the comment tab layout**
* **Send notification to publisher when someone has commented on their research paper**

**User story 3:** As a research scholar, I want to ask doubts, in the form of comments on any research paper

**Sub-Tasks:**

* **Configure the database, with the table schema to store the comments**
* **Update notifications page when a comment is added**
* **Push notification to the user's email when they receive a reply to their comment.**
* **Set user permissions to allow to add/edit/delete a comment**
* **Design a tab below the research paper to view the reviews/feedbacks on the paper**
* **Create SQL scripts to store the comments in the database**

**User story 4:** As an admin, I want to delete comments based on the content

**Sub-Tasks:**

* **Allow access to admin to delete/hide others comments**
* **Host the backend in a cloud application like Heroku**
* **Cut SVG icons and images**
* **Setup GitHub project repo**
* **Create a CI/CD pipeline using GitHub Actions**
* **Implement review page using HTML/CSS/JS**

**User story 5:** As a user, I want to upvote a comment on a repository

**Sub-Tasks:**

* **Update the comments tab, when a new comment is added**
* **Add SVG icon for upvote**
* **Create a REST API to update the value of the upvote on the comments**
* **Provide a button to upvote a comment**
* **Create SQL scripts to store the comments status [no. of upvotes] in the database**
* **Implement comments tab using HTML/CSS/JS**

**Epic:** **UI Customization**

**User story 1: As a user I want an option to switch from dark mode and light mode to control the blue light emitted from the screen.**

**Sub-Task:**

* **Design the button layout**
* **Provide a button that helps the users  to toggle between the different modes.**
* **Implement upload page using HTML/CSS/JS**

**User story 2:** **As a user I would like to change the size of the text to read articles more conveniently.**

**Sub-Task:**

* **Provide a button which redirects to the font-family page.**
* **Design a text control page that lets the users view the size entered.**
* **Provide a textbox along with the dropdown button that reads the desired text size input from the user.**
* **Design a dropdown button showcasing all the text sizes.**
* **Implement it using HTML/CSS/JS**

**User story 3: As a user I would like to change the font style from a given family of fonts.**

**Sub-Task:**

* **Provide a button which redirects to the font-family page.**
* **Design a Font family page that lets the users preview the font selected.**
* **Upload a set of predefined fonts from google fonts.**
* **Design a page that displays all the given fonts and   lets the users choose from it.**
* **Implement it using HTML/CSS/JS.**

**User story 4: As a user I would like to change the reading layout between a list view and a grid view.**

**Sub-Task:**

* **Provide a view button which pops out the view settings bar**
* **Design a view bar with a dropdown button to select the view types.**
* **Design a Save button to save the finalised Font.**
* **Implement it using HTML/CSS/JS.**

**Epic:** **Search and sort**

**User story 1: As a registered/new user of the repository, I would like to view relevant documents by searching for keywords present in the titles.**

**Sub-Task:**

* **Provide a “Search Options” button, which reveals a dropdown menu on clicking.**
* **Design the dropdown menu where “Search by keywords” is an option.**
* **Implement an algorithm that, when given one or more keywords, can efficiently return the list of all files with the keywords in the title.**
* **Create a page with a search bar on top saying "Search by Keywords"**
* **Write a script such that entering the keywords in the search bar lists all publications with the keywords in their title**
* **Create the above page using HTML/CSS/JS**

**User story 2:** **As a registered/new user of the repository, I would like to search for publications by a specific author**

**Sub-Task:**

* **Provide a “Search Options” button, which reveals a dropdown menu on clicking.**
* **Add a “Search by Author” to the Search Options dropdown menu.**
* **Implement an algorithm that takes in the author’s name and returns all publications by that author.**
* **Create a page with a search bar on top saying "Search Author's name"**
* **Write a script such that entering the name of an author in the bar lists all publications by that author.**
* **Create the above page using HTML/CSS/JS**

**User story 3: As a registered/new user of the repository, I would like to filter the publications based on publishing date**

**Sub-Task:**

* **Provide a “Search Options” button, which reveals a dropdown menu on clicking.**
* **Add a “Search Filters” option to the Search Options dropdown Menu.**
* **Implement an algorithm that takes in two Dates and returns all the publications published between those dates.**
* **Add a “Search Filters” option to the Search Options dropdown Menu.**
* **Create a script such that entering the Start Date and End Date outputs list of all documents published in that range.**
* **Create the above page using HTML/CSS/JS**

**User story 4**: **As a registered/new user of the repository, I would like to sort the publications based on publishing date**

**Sub-Task:**

* **Add a button next to each column to sort based on it.**
* **Create the button using HTML/CSS/JS**
* **Write a script which alternately sorts in ascending and descending order each time button is pressed.**
* **Implement a sorting algorithm that sorts all publications using a particular parameter.**
* **Write a script which calls the sorting algorithm when the button is clicked.**

**Epic:** **Notifications to the publisher/user**

**User story 1**: **As a publisher/faculty, I want to receive notification when my paper is added to the repository successfully**

**Sub-Task:**

* **Implement notification page using HTML/CSS/JS**
* **Refresh/Update notification when a notification action is triggered**
* **Redirect to the profile page showing all your papers**
* **Create scripts to ensure that all the details and files provided are safely secured in the database**
* **Push notifications to the publisher's email when he/she has successfully uploaded a paper to the repository**
* **Provide an icon that indicates there are new notifications**

**User story 2**: **As a publisher/faculty, I want to receive notification when someone has commented on my paper**

**Sub-Task:**

* **Create a REST API to push notifications to either email/telegram/in-app**
* **Configure FormSpree to receive notification when a form is filled in the app**
* **Cut SVG Icons**
* **Redirect to a link in a notification, when clicked**
* **Implement notification page using HTML/CSS/JS**
* **Push notification to publisher, when a comment is added to his/her paper**

**User story 3**: **As a publisher/faculty, I want access to allow or block notifications from the website.**

**Sub-Task:**

* **Update the database, of the user preferences**
* **Update notification settings tab, on-change**
* **Create a button to allow/block in-app notifications**
* **Create a button to view notification as a popup**
* **Create web service REST API for user resource**
* **Push notification to the user when the app is updated or under maintenance**

**User story 4**: **As a user/research scholar, I want to get notification when someone has replied to my comment.**

**Sub-Task:**

* **Ensure that the links and buttons work properly as they should**
* **Design a tab to view notifications as pop-up**
* **Push notification to the user when someone has replied to their comment [sub-comment]**
* **Create Unit Tests to check that in-app and email/telegram notifications work properly**
* **Redirect to a link in the notification, when clicked**
* **Provide an icon that indicates there are new notifications**

**User story 5**: **As a user, I want to turn on/off notifications for my comments.**

**Sub-Task:**

* **Create unit tests to check the working of buttons**
* **Provide a button that allows/block notifications of comments and their replies**
* **Update the notifications page, when a notification action is triggered**
* **Allow users to set their custom notification settings**
* **Update the database, of the user preferences**
* **Design the notification tab layout**

**Epic:** **Registration and Authentication**

**User story 1**: As a registered member of the repository, I should be able to login to my account in order to access the functionalities of the repository

**Sub-Task:**

* + **Develop the website using HTML/CSS and JS**
  + **Design a layout of the page providing a text box one to enter the email**
  + **Design a text box to enter the password**
  + **Create scripts to ensure that all the details and files provided are safely secured in the database**
  + **Design a button which sends the information entered by the user to be validated**

**User story 2**: As a new user to the repository, I should be able to register myself in order to become a member of the repository.

**Sub-Task:**

* **Implement update page HTML/CSS/JS**
* **Design a layout of the page providing a text box, one to enter the name and one to enter the email address. third one for password and finally the last one for confirming password.**
* **Create scripts to ensure that all the details and files provided are safely secured in the database**
* **Design two additional text boxes one for password and finally the last one for confirming password.**
* **Upon successful verification write scripts to create a new record in the database**
* **Design a button which sends a confirmation email to validate the email entered by the user**

**User story 3**: As a registered member of the repository, I should be able to reset my password in order to recover my account and not lose any information.

**Sub-Task:**

* **Design a button which redirects the user to the forgot password page**
* **Develop the website using HTML/CSS and JS**
* **Create scripts to replace the newly created password in place of the old in the data base.**
* **Design a text box for the user to enter the email id associated to the account**
* **Design a button which upon being clicked sends an email to the one entered by the user which the user can then use to reset his password.**

**User story 4**: As a user of Google’s services, I should be able to seamlessly login/signup with my pre-existing Google account.

**Sub-Task:**

* **Develop using HTML,CSS and JS**
* **Create Authorisation credentials to access Google OAuth API**
* **Load the Google platform Library**
* **Get Profile Information**
* **Design a Google Sign In button**

**Epic:** **Feedback**

**User story 1**: As a registered user of the repository, I should be able to flag a research paper if I believe it has content which I deem is inappropriate, irrelevant, spam/misleading in order to preserve the integrity of the repository.

**Sub-Task:**

* **Develop the website using HTML/CSS and JS**
* Design an icon next to each research paper listed in the repository which upon being clicked will present a list of options.
* Write scripts to create a new issue and update the database.
* Design a dropdown list with the options which can be clicked which will in turn send a message to the administrator of the repository.

**User story 2**: As a registered user of the repository, I should be able to flag any user if I believe he/she has posted content which I deem is malicious, spreading hatred, spam/misleading in order to preserve the integrity of the repository.

**Sub-Task:**

* Develop the website using HTML/CSS and JS
* Design an icon next to each user in the repository which upon being clicked will present a list of options.
* Write scripts to create a new issue and update the database.
* Design a dropdown list with the options which can be clicked which will in turn send a message to the administrator of the repository.

**User story 3**: As a registered user of the repository, I should be able to report any bugs that I may have encountered with the website or provide suggestions so as to improve its functionality and improve the user experience for myself and other users of the repository.

**Sub-Task:**

* Design a button which redirects the user to the “Feedback” page
* **Develop the website using HTML/CSS and JS**
* Design a button to upload pictures/screenshots (if any)
* Design a textbox for the description of the issue.
* Design a Textbox for the title of the issue encountered.

**INVISION LINK**

[**https://rprepo.invisionapp.com/freehand/User-Stories-p2RCBSFEO**](https://rprepo.invisionapp.com/freehand/User-Stories-p2RCBSFEO)