

Proj-rec, F24-06

Xinghan, Michael J, Skanda, Victor, Chris L, Edward

Motivation:

Each year, a lot of UCSB seniors choose to continue their academic journey by going to a Master's, PhD, or another professional program to leverage their specialized experience in a certain field. In that case, they usually need to apply to a lot of programs, thus requesting recommendation letters from faculties. For both faculties and students, they need to keep track of and manage their requests for recommendation letters and their status.

Specifications and Deliverables for Our Project

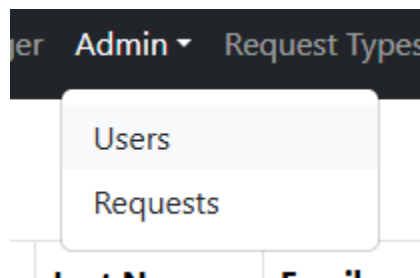
To spearhead this process, we developed a web application feature where students can submit requests to professors, and professors can manage students's requests. There are information pages for professors and students and the database and APIs will make the process much more efficient and smooth for both parties in the process. We also use frontend technologies to make it more user-friendly.

Release Notes:

What's New:

1. Users

- There are now 2 more roles for users. Other than admin and user roles, there are the professor and student roles.
- Admin, Professor, and Student roles can now be toggled by admin users.
- The toggles can be found at the users table at /admin/users.



- There are three bottoms, labelled Toggle Admin, Toggle Professor, and Toggle Student for each user for Admins to use.
- A user's roles are also displayed on this user's table through 3 booleans columns, admin, professor, and student.

2. Recommendation Request Form

- Form component for creating and editing recommendation requests
- Based on the POST method for the recommendation request controller
- Allows users to create recommendation requests for professors.
- There's a dropdown for request types
 - This is currently a placeholder, but it would show the request types added in the database

Professor Email Request Type

--Select an Option --

Details

3. Delete all legacy and non-related features to Request

- There are several features in our web application, including Restaurant, UCSBDate, UCSB dining commons, etc.
 - Key Features
 - 70+ files deleted non-related to Request
 - Backend APIs, databases, entities, etc.
 - Frontend pages, storybook, navigation bar
 - Forms and utility functions
 - Related test files including Integration and end-to-end test
- In the process of deleting all 70+ files, some tests we deleted covered some API helpers or request functionalities that we kept so we need to write additional tests to ensure 100% test coverage. For the frontend, we need to rearrange the layout of our navigation bar and some other utilities.

4. Request Types

- For our application, we want admin and professors the ability to manage different types of requests. This improvement simplifies categorizing and managing request types for improved workflow.
- Default request types are as follows:
 - CS Department BS/MS program
 - Scholarship or Fellowship
 - MS program (other than CS Dept BS/MS)
 - PhD program
 - Other

5. Admin Requests Page

We want the admin to be able to see all of the recommendation requests in the database. Right now there is a placeholder titled "All Recommendation Requests".

6. Professor

We want professors to be able to see all of the recommendation requests that have been sent to them. Right now we added a professor tab in the navbar with links to pending requests, completed requests, and statistics

page. Each page is a placeholder that renders the corresponding title. Only users that have the professor role can navigate to these pages and see the professor tab on the navbar.

7. Pending Requests Page

We want professors to be able to see all of the pending requests and be able to accept and deny requests, moving them to completed requests. Right now, the frontend is nearly fully implemented through the Table but there is no backend implementation.

ID	Student Email	Status	Request Date	Accept	Deny
1	student1@example.com	Pending	2024-11-01T10:00:00	Accept	Deny
2	student2@example.com	Pending	2024-11-02T12:00:00	Accept	Deny
3	student3@example.com	Accepted	2024-11-03T14:00:00		

Work in Progress:

1. Student profile

- Add a new page to frontend/src/main/pages for the Student Profile Page
- Student logs in and can access their profile
 - Can see all their existing requests and their associated status
 - Can create, read, update, delete, their recommendation requests

2. Request Types

- In its current state, the request type management feature needs three frontend capabilities, which are shown in more detail below.
- Key needs:
 - The ability to create new Request types like shown below
 - The ability for admin/professor to edit current request types
 - The ability to monitor all current request types

Create New Request Type

Request Type

Create Cancel

3. Recommendation Request Create Page

Currently, though backend support exists for recommendation requests, the create/edit page to house the form component for these requests is currently a work in progress.

4. Admin Requests Page

Table component to see all recommendation requests is WIP. Waiting for the backend for recommendation requests to be completed.

5. Pending Requests Page

Page component and backend implementation are WIP. Completed requests page also needs to be completed for a full and thorough implementation.

6. Professor

Table components for the professor pages are WIP. Waiting for the backend for recommendation requests to be completed.

Dev. Info: Additional Information

1. Recommendation Requests

The backend for recommendation requests is only missing endpoints for searching by status.

Currently the following api endpoints have been implemented:

RecommendationRequest			^
GET	/api/recommendationrequest	Get a single recommendation request	▼
PUT	/api/recommendationrequest	Update a single recommendation request	▼
DELETE	/api/recommendationrequest	Delete a recommendation request	▼
POST	/api/recommendationrequest/post	Create a new recommendation request	▼
GET	/api/recommendationrequest/requester/{userId}	Get all recommendation requests by Requester's user id	▼
GET	/api/recommendationrequest/professor/{userId}	Get all recommendation requests by Professor's user id	▼
GET	/api/recommendationrequest/all	List all recommendation requests	▼

For the backend, there are Many-To-One mappings from Recommendation Requests to Users, as well as to Request Types. These can be observed in the entity file for Recommendation Requests. The api endpoints use the unique ids of users and requests types to link them to recommendation requests. There ids are stored together with the requests in the database.

Key needs:

- retrieve by status controller endpoint
- Frontend Completed Page, for professors to view to their requests that have a status of “accepted” or “denied”
- Frontend Pending Page, for professors to view any requests directed to them that have a status of “pending”
- Frontend Statistics Page, or professors to display the number of completed requests and number of requests that still need reviewing.
- Another implementation of the Statistics Page could be to display max, min, and average response times a professor has for requests, (compare date of submission to completion date)

2. Request Types

Request types are fully working and implemented in the backend, but are missing some key frontend features. These missing components are detailed below.

Key needs:

- A frontend create page that allows admins/professors to add any new request type
- A frontend edit page that allows admins/professors to edit any request type
- A frontend index page that allows admins/professors to see all the request types

The majority of the work has already been done for the create page on branch `cl-reqtypes-create` and only minor fixes need to be done for the create page to be up and running.

Shown below are all the backend features that have been implemented for request types.

Request Types			^
GET	/api/request-types	Get a single request type	▼
PUT	/api/request-types	Update a single request type	▼
DELETE	/api/request-types	Delete a Request Type	▼
POST	/api/request-types/post	Create a new request type	▼
GET	/api/request-types/all	List all request types	▼