

Ph.D. Application Review System

Final Report

Fall 2023

CSCE 606

Texas A&M University

Ruby off the Rails

Meetansh Gupta

Shreyas Kasetty

Brian Munoz

Tanay Patankar

Francesco Romano

Ruofei Xu

1. Project Summary

The project focused on streamlining the PhD application review process by developing an application that replaces manual data entry into a Google Sheet. The stakeholders of this project are all faculty in the department of computer science and engineering at Texas A&M University. The main objective of the application was to facilitate the addition of candidate information and enable faculty members to review these applications. The application's workflow started with a committee chair conducting an initial review of a candidate and then forming a committee of faculty members to further assess the application. Reviews consisted of a textual analysis, a numeric score ranging from 1 to 5, and whether the candidate should get an assistantship. To enhance user experience, additional features were incorporated into the system, such as a search page for faculty members to find candidates based on specific criteria and a user-friendly dashboard displaying unreviewed assigned candidates for prioritized review, in-progress reviews, and completed reviews.

Throughout the project, several iterations and updates were made to the application. The user interface was refined with features like login integration through TAMU CAS and a separate admin login. We also added functionality for adding and editing reviews and creating faculty committees for application assessment. Further improvements included the development of a secure API for database access, a responsive user interface adaptable to different device screens, and a more robust backend with enhanced data security features. The final stages of the project focused on refining the user experience, increasing the flexibility and security of the database system, and ensuring comprehensive test coverage for reliability and performance. The team's

collaborative efforts resulted in a user-centric, secure, and efficient application for managing the complex process of Ph.D. application review.

2. User Stories

See Appendix

3. Team Members

Scrum Master(s): Francesco Romano Iterations 1- 3 and Brian Munoz: Iterations 3-5

Product Owner: Tanay Patankar

Front End Specialists: Meetansh Gupta, Shreyas Kasetty

Database Gurus: Francesco Romano, Brian Munoz

Back End Specialists: Ruofei Xu, Tanay Patankar, Meetansh Gupta, Shreyas Kasetty

4. Scrum Iteration Summary

<u>Iteration 0:</u>	Introduction to the project and role assignments
<u>Iteration 1:</u>	<ul style="list-style-type: none">• App created and TAMU CAS login functionality• A database was designed and created to store reviews• Functionality to convert the csv from WebAdmit to a json• UI for the login page and homepage <p>Total points: 60</p>

<u>Iteration 2:</u>	<ul style="list-style-type: none"> ● An independent FEPRA protected database with the functionality to import the application data was created. ● API created to search the independent database with custom fields and values ● Functionality to track users and for an admin to login ● Appropriate tests were added for the app <p>Total Points: 40</p>
<u>Iteration 3:</u>	<ul style="list-style-type: none"> ● Update to the search functionality and its integration on homepage ● Updates to the homepage and login page UI with dynamic elements ● Functionality for faculty to add and edit reviews with text, and a rating ● Database updated with applicants table and with user authorization ● Security added to the search API ● Bug fixes and appropriate tests ● Deploying and Testing instructions on Github <p>Total points: 85</p>
<u>Iteration 4:</u>	<ul style="list-style-type: none"> ● Admin dashboard to manage CRUDs with admin-only restriction ● JWT authentication to the API and other authorization features ● Database reset functionality ● Updates to review add/edit features ● Tasks list for faculty to view assigned, in progress, completed reviews ● Major upgrades to the UI across the app ● Database flexibility highly increased ● Bug fixes and appropriate tests <p>Total points: 65</p>

<u>Iteration 5:</u>	<ul style="list-style-type: none"> • UI for the database with dynamic elements • Added routes and methods for handling GUI inputs • Implemented and refactored database with more robustness • Added reviews functionality on the front-end • Feature for committee chair to assign reviews to faculty • Dashboard UI for chair to manage and assign reviews • Refactoring code, bug fixes and appropriate tests <p>Total points: 50</p>
---------------------	---

Points Breakdown Table for All Iterations:

	Francesco	Tanay	Meetansh	Ruofei	Brian	Shreyas	Sum
i1	10	10	10	10	10	10	60
i2	7	8	4	8	5	8	40
i3	20	16	19	7	8	15	85
i4	12	12	10	9	6	16	65
i5	3	4	5	14	20	4	50
							300

BDD/TDD Process

For the BDD (Behavior-Driven Development) process, we focused on defining the PhD application review system behaviors from the client's perspective. We had meetings with the client and then we wrote the user stories and we wrote the scenarios referring to them. Then we automated the scenarios by the tests in Cucumber and Rspec. And we

Implemented code to make tests pass. By these steps, it ensures that the system behaves as intended from a user's perspective.

For the TDD (Test-Driven Development) process, we wrote tests before the code and followed the Red-Green-Refactor cycle. When we had a failing test we wrote the code to make the test pass. Every time we refactored the code, we made sure that the tests were still passed.

BDD and TDD bring numerous advantages to software development. They enhance code quality, ensure comprehensive test coverage, foster collaboration among team members, expedite debugging, and provide living documentation for the system's behavior, which aids in long-term maintenance and evolution.

Issues on Heroku Deployment

- Heroku Add-ons: We need to enable Postgres in Heroku App's add-ons as the database for our project.
- Store the Config Vars in Heroku App: We store credentials and keys in the Heroku App's Config Vars so that we can use the credentials for the JWT token, Google Auth, etc on the Heroku App.
- Heroku console problem: If the developer leaves the Heroku App's rails console without stopping the console, the developer must stop the console service and restart the rails console on the Heroku App.

Customer Meetings

Iteration 0 Meeting 1:

Date: September 5th

Time: 1 pm-2 pm

Place: PETR 214

Meeting Description:

Discuss the goals for our application, which include:

- allowing faculty to review applications and generate a comprehensive report of all reviews.
- adding candidate information to the system and a committee chair conducting an initial review.
- Assigning faculty to review the application and provide a text review and a numeric score from 1-5.
- Adding a search page that enables faculty to find candidates based on specific criteria
- Create a user-friendly home page that displays unreviewed assigned candidates as high-priority
- The candidate add page that can import a candidate's data from a CSV file or other formats to be determined.

Iteration 1 Meeting 1:

Date: September 18th

Time: 2 pm-3 pm

Place: PETR 214

Meeting Description:

Important features:

1. Creation of a white list for login access.
2. Filter capabilities should include the following logic as well: X and (A or B or C) queries. Queried structure
3. Database backup feature
4. Implement CSV import functionality

Iteration 1 Meeting 2:

Date: September 29th

Time: 2 pm-3 pm

Place: PETR 214

Meeting Description:

Contact CAS staff for a whitelist

UI Stuff:

Sidebar Icons to be smaller

Search bar good

No need for any search results on the homepage

Dashboard should be a personalization thingy:

Faculty dashboard -

shortlisted or interested in or reviewed applications

able to create kinda like a wishlist of applications

HotCRP: Need functionalities similar to this

Chair: the ability to assign applications to member

 Apply a particular application to a particular person

 Can send an email to the system once applications are assigned,

 NO AUTO EMAILS TO BE SENT - to be triggered by chair only

Assignments: Make suggestions to which 3-4 professors should this application go to.

But the ultimate decision is in the hands of the chair

 The chair should be able to change the reviewers

Emails: Content needs to be conferable: Able to insert text and replace fields

Notifications Page: Not required!

Iteration 2 Meeting 1:

Date: October 13

Time: 3:30 pm-4:00 pm

Place: PETR 314

Meeting Description:

Presented application structure as a comprehensive list of user stories

Different layout for the tasks - Use one row instead of large squares

Task status need not be explicit

Wants to have a script that automatically scrolls down to the SOP when on an application PDF.

Iteration 3 Meeting 1:

Date: October 20

Time: 2:00pm-3:00pm

Place: PETR 314

Meeting Description:

- After a search capture the total number of results at the bottom of the search results.
- Have aliases for the field names so that search queries are easier.
- Have authentication for the API

Iteration 3 Meeting 2:

Iteration 3 Demo Meeting 2

Date: October 27

Time: 2:00 pm-3:00 pm

Place: PETR 314

Meeting Summary:

Main Application:

- hyper link to whole field
- use 5 radio buttons for rating instead of a drop down

- Hover over "Reviewed By" reveals a pop-up showing reviews from others
- Load PDF in browser
- Review box should not be embedded in HTML
 - It should be floating, so the user doesn't have to scroll all the way down to write and submit their review
- Add link to go to database app
- -TTL for token: not needed if token is tied to session activity
- Don't restrict IP addresses
- Add GAT/GAR checkbox D
- Add Local Tag functionality
 - Can use tag to search database

Database Application

- Add interface to change fields on database app
- Add reset functionality
 - Brand new state
 - no schema, no entries, preserve users
- Cannot reupload multiple CAS IDs

Iteration 4 Meeting 1:

Date: Nov 3

Time: 2:00 pm-3:00 pm

Place: PETR 314

Meeting Summary:

- Notify user of duplicates (Now obsolete, we are redesigning the database)
- Fix the bug when you refresh _savedata in the Database App
- Don't repeat the rating
- Two separate checkboxes for GAT and GAR
- Implement JWT for the database system API

Iteration 4 Meeting 2:

Date: Nov 10

Time: 2:00 pm-3:00 pm

Place: PETR 314

Meeting Summary:

- Investigate the use of TAMU PC in Peterson to deploy app and database
 - See what we need and report back to TAMU IT
- Make sure tags are searchable
- Have a way to express local and global tags
- Restrict the admin page to the admin role user

Iteration 5 Meeting 1 (Last):

Date: Nov 17

Time: 2:00 pm-3:00 pm

Place: PETR 314

Meeting Summary:

- Randomly and equally distribute work to all committee members
 - At least 2 reviewers
 - Make Dr. Nguyen's life easier
- Include the name of the Applicant, keep the ID
- Make DB schema explicit
- Load example schema
 - Try CSV for this year
- Deploy project on TAMU machine
- Get the PDF viewer fully working
- Add/finish review User Interface

Challenges about Github

- Merge Conflicts: When multiple team members are working on the same file or section of code, conflicts occur during the merge process. We usually resolve these conflicts in Github editor, and sometimes we locally resolve the conflicts on VSCode.
- Branch Management: Because we have 6 contributors, branch management becomes crucial. We created new branches from the main branch before we worked on a new feature, then we merged the branch into main. In this way, we can handle multiple features and bug fixes concurrently without disrupting the main codebase. This approach ensures that the main branch always remains stable and deployable.

Tools We Used

- SimpleCov: SimpleCov is a Ruby gem used for code coverage analysis. It helps us measure test coverage in our codebase, identifying areas lacking proper testing. We used SimpleCov to generate detailed reports, making it easier to improve test coverage and overall code quality. It's a valuable tool for ensuring software reliability and reducing the risk of undetected bugs.
- Postman: Postman is a tool offering a comprehensive platform for API testing and development. we used Postman efficiently test your API endpoints, automate testing processes, collaborate with team members on API documentation. It helps us catch bugs early in development and streamline the testing and debugging process.
- PgAdmin: PgAdmin is an open-source management tool designed for PostgreSQL databases. We used PgAdmin to simplify the tasks like database creation, query execution, and performance optimization.

Configuration Management

- Our approach to configuration management consisted of the following:
 - 1.) When working on a new feature or bug fix, a developer would sync their Version Control management software (GitHub) with the main branch.
 - 2.) The developer would then make a new branch from Main, naming it according to their work
 - 3.) After the feature has been implemented and tested, a developer would make a pull request
 - a.) This pull request required 2 reviews from the team and their approval

- 4.) Once approved, the new code within their branch would be merged into the Main branch and deployed automatically to Heroku
- a.) If there are any errors during deployment, the team will get an automated email from Heroku notifying them
- With regard to spikes the work was evenly distributed throughout the semester for the most part. We did have to work a little bit more towards the end of the project to try and meet the customer's requirements. Overall, there were minimal spikes except near the end of the project.
 - In the end we had 44 branches for our main application and 14 branches for our database application.

About our Repository and Code Set-up/Deployment:

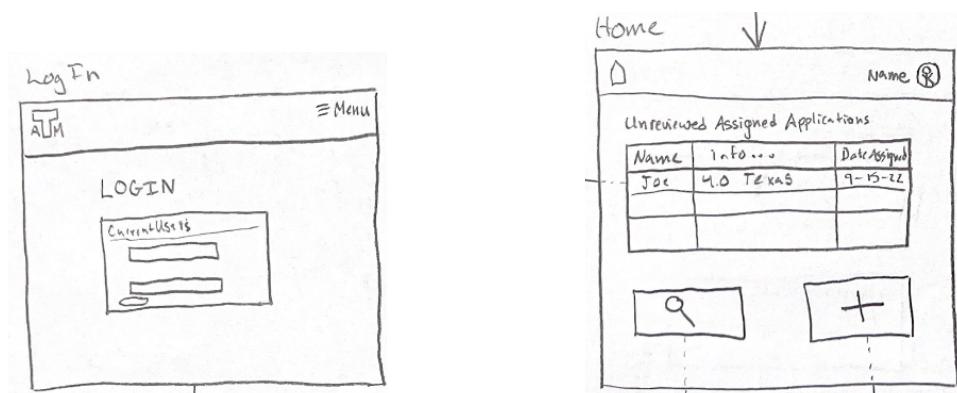
Please visit: [RubyOffTheRailsTAMU/PhD-Application-Review-System \(github.com\)](https://RubyOffTheRailsTAMU/PhD-Application-Review-System (github.com))

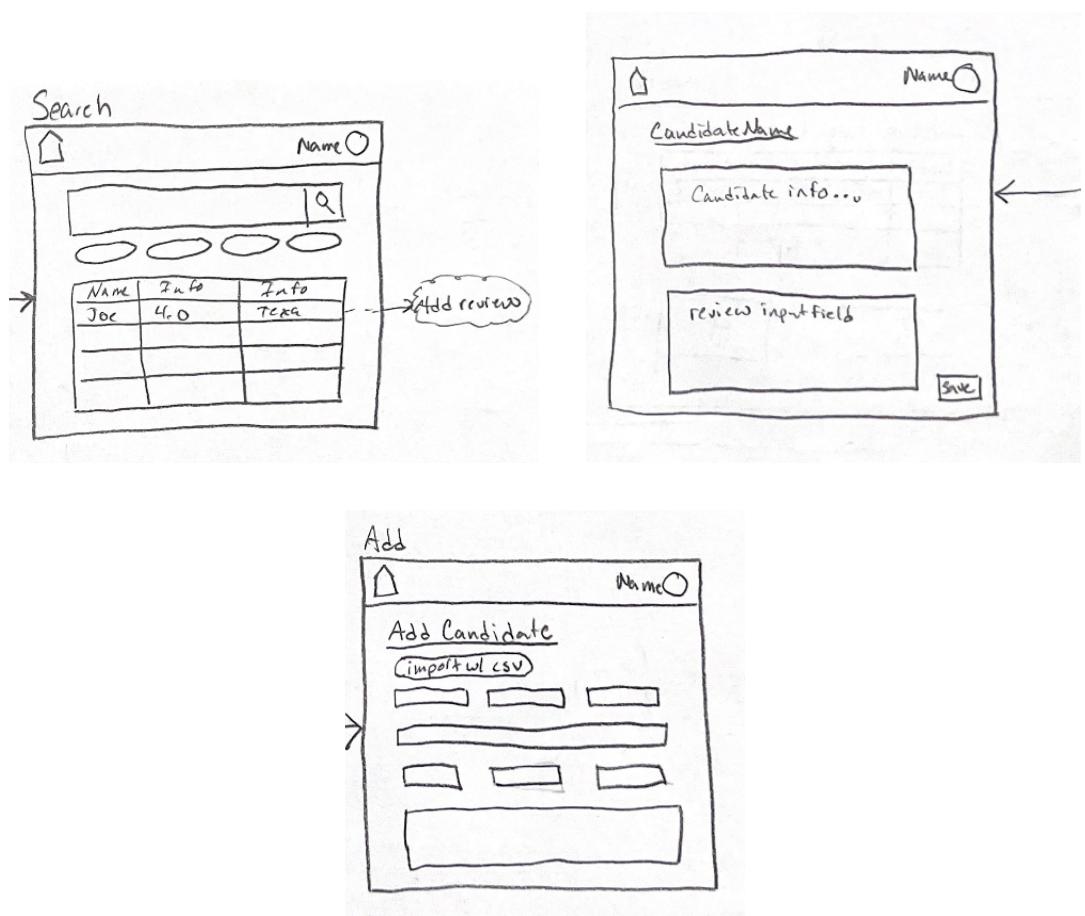
Appendix

User Stories

Iteration 0 (No Stories)

Lo-Fi Mockups:





Iteration 1

[Tanay and Shreyas; points: 10]

Feature: Login with CAS

As a user

So that I can access the PHD Review Application\

I want to be able to log in through TAMU CAS

[Tanay; points: 5]

Feature: Integration Test for login

As a Developer,

so that i can test the functionality of google oauth login

i want to run rspec tests that cover this functionality and give a coverage report

[Brian and Francesco; points: 10]

Feature: Create User database table

As a developer

So that users of the System can have a personalized experience and appropriate
privileges

I want to make a table in the database with User-specific information

[Francesco; points: 5]

Feature: Create Database relations diagram

As a developer

So my team can implement the application

I want to design the database and its relations

[Brian; points: 5]

Feature: Create Review database table

As a developer

So that users of the System can review applicants

I want to make a table in the database that will hold the relevant information from
submitted applicants

[Ruofei; points: 5]

Feature: Cucumber test and Rspec test

As a developer

So that I can test the features of the project

I want to be able to run tests by Cucumber and Rspec and view coverage the report

by simplecov

[Ruofei; points: 5]

Feature: convert CSV to JSON

As a developer

So that I can easily get data from JSON

I want to be able to convert the application CSV file that I get from WebAdMIT API.

[Shreyas; points: 5]

Feature: Dockerizing the Ph.D. Application Review System

As a developer

I want to dockerize the Ph.D. Application Review System

So that the application can be deployed and environment setup can be handled

easily

[Meetansh; points: 5]

Feature: Homepage UI

As a user

So that I can access all app functionalities

I should be redirected to and able to see the homepage.

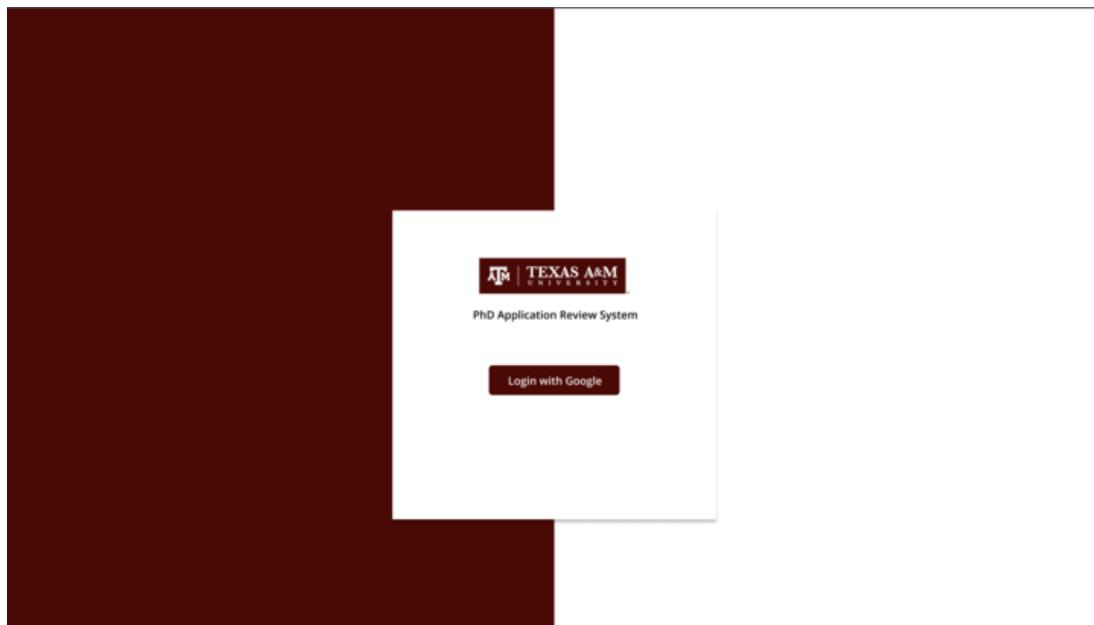
[Meetansh; points: 5]

Feature: Logout Button

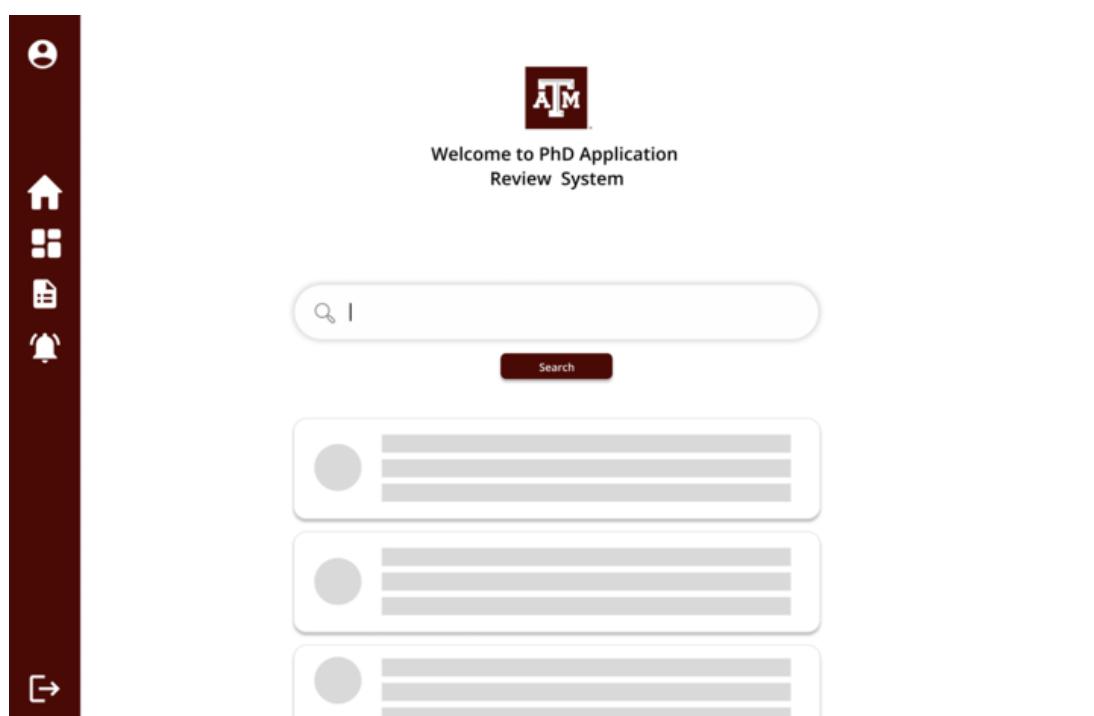
As a user

So that I can logout of the app

I should be able to click the logout button and go to the login page



Login Page



Home Page

Iteration 2

Feature: Search functionality

Francesco 6

As a user

So that I can find a particular applicant

I want to specify the criteria for searching the applicant

And when i click on an applicant

I want to be redirected to the applicant review page

Feature: Tasks

Shreyas 8

Feature: Show committee members assigned reviews

As a committee member

So that the applicant gets reviewed promptly

I want to see notifications about applicants I need to review

And when i click on an applicant

I want to be redirected to the applicant review page

Feature: Import data page

Brian 5

As a user

So that i can add new applicants into the system

I want to be able to upload a csv file into the system

Feature: Applicant Independent Database

Ruofei 8

As a developer

So that the Ferpa data is protected

I want to create an independent database for applicants

Feature: Users table for auth users

Tanay 4

As a developer

So that i can track users through the system

I want to have a table of users

Admin login

Tanay 4

As an Admin

So that I can access restricted app features

I want to be able to log in as an admin

And be redirected to and able to see the homepage.

User stories list

Francesco 1

As the client

So that i can ensure all my specifications are met

I want a comprehensive list of all features and uses of the app

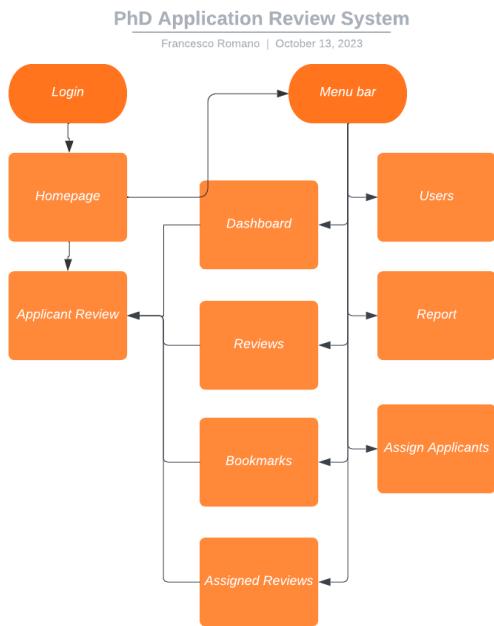
Cucumber Tests for applicants and search

Meetansh 4

As a developer

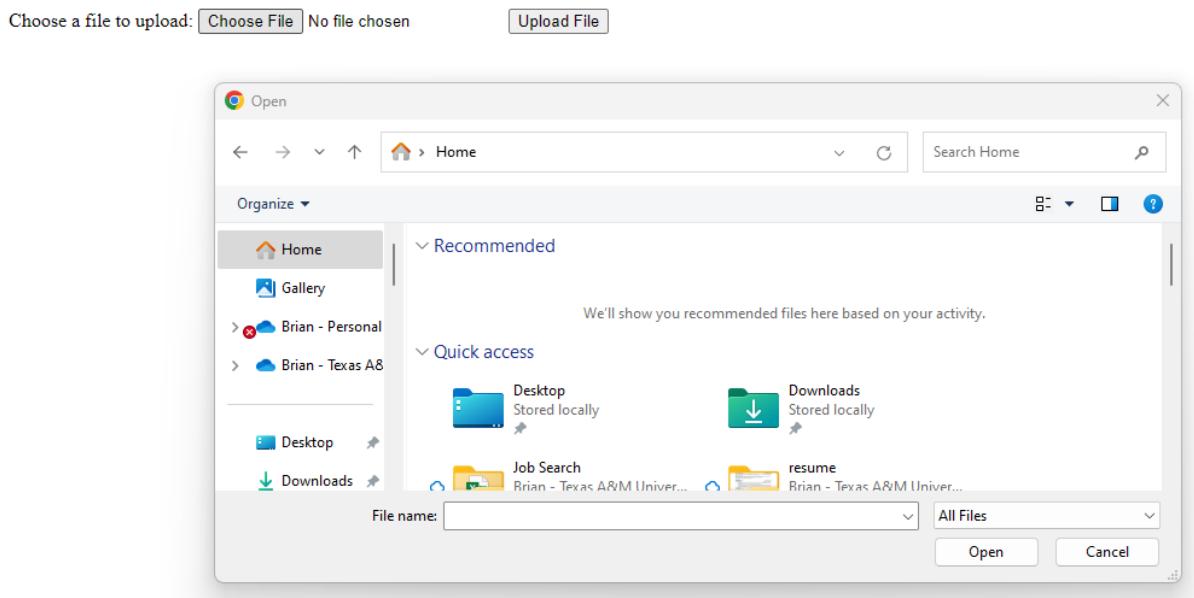
So that i can ensure my code is quality

I want to write tests to check applicants and search functionality



Application Diagram

Database System:



UI to Upload Files w/ System Dialog

Test Report:

This report will self-destruct in a day. Keep your future reports forever.

6 PASSED	1 UNDEFINED	2 FAILED
67% passed 9 executed	5 hours ago Last Run	5.87 seconds Duration
 linux	 ruby 3.2.2	 cucumber-ruby 8.0.0
Delete Report		
<ul style="list-style-type: none">>  features/addreview.feature>  features/csvtojson.feature>  features/login.feature>  features/newreview.feature>  features/reviews.feature>  features/searchapplicant.feature		

© 2023 SmartBear Software. All Rights Reserved.
[Privacy](#) | [Terms of use](#)



Cucumber Coverage Report

Generated about 5 hours ago

All Files (64.87% covered at 17.68 hits/line)

29 files in total.
316 relevant lines, 205 lines covered and 111 lines missed. (64.87%)

File	% covered	Lines	Relevant Lines	Lines covered	Lines missed	Avg. Hits / Line
app/controllers/applicants_controller.rb	5.06 %	202	79	4	75	0.10
app/controllers/reviews_controller.rb	38.24 %	71	34	13	21	0.76
app/controllers/users_controller.rb	60.00 %	72	35	21	14	1.00
app/controllers/sessions_controller.rb	96.00 %	52	25	24	1	2.00
app/channels/application_cable/channel.rb	100.00 %	4	2	2	0	2.00
app/channels/application_cable/connection.rb	100.00 %	4	2	2	0	2.00
app/controllers/application_controller.rb	100.00 %	22	15	15	0	7.47
app/controllers/homepage_controller.rb	100.00 %	2	1	1	0	2.00
app/controllers/searches_controller.rb	100.00 %	15	11	11	0	2.18
app/controllers/welcome_controller.rb	100.00 %	14	10	10	0	2.60
app/controllers/xlsx_controller.rb	100.00 %	139	71	71	0	73.72
app/helpers/application_helper.rb	100.00 %	2	1	1	0	2.00
app/helpers/csv_helper.rb	100.00 %	2	1	1	0	2.00
app/helpers/reviews_helper.rb	100.00 %	2	1	1	0	2.00
app/helpers/searches_helper.rb	100.00 %	2	1	1	0	2.00
app/helpers/sessions_helper.rb	100.00 %	2	1	1	0	2.00
app/helpers/users_helper.rb	100.00 %	2	1	1	0	2.00
app/helpers/welcome_helper.rb	100.00 %	2	1	1	0	2.00
app/helpers/xlsx_helper.rb	100.00 %	2	1	1	0	2.00
app/jobs/application_job.rb	100.00 %	7	1	1	0	2.00
app/mailers/application_mailer.rb	100.00 %	4	3	3	0	2.00
app/models/applicant.rb	100.00 %	6	5	5	0	2.00
app/models/application_ielt.rb	100.00 %	3	2	2	0	2.00
app/models/application_record.rb	100.00 %	3	2	2	0	2.00
app/models/gre.rb	100.00 %	3	2	2	0	2.00
app/models/review.rb	100.00 %	4	1	1	0	2.00
app/models/school.rb	100.00 %	3	2	2	0	2.00
app/models/toefl.rb	100.00 %	3	2	2	0	2.00
app/models/user.rb	100.00 %	4	3	3	0	2.00

Showing 1 to 29 of 29 entries

Generated by simplecov v0.22.0 and simplecov-html v0.12.3
using Cucumber Features, RSpec

Rspec Coverage Report

Iteration 3

Feature: Search functionality update

Francesco 10

As a user

So that I can find a particular applicant

I want to specify the criteria using a common phrase

And those should be seen on the drop down list

And the number of results should be shown

And when i click on the applicant

I should be redirected to the review page

Feature: Search Integration

Francesco 10

As a user so I can search from the home page

I should be able to use the search on the homepage

And be redirected to the search page

Feature: Home page UI

Shreyas 5

As a user so I can resize the page dynamically

I should be able to change the window size

And the page should look fine

Feature: Robust API Security for Application Database Access

Shreyas 10

As a developer,

In order to ensure safe and secure access to my application's database,

I want to design and implement an API with strong security measures.

Feature: Add Reviews to the Application

Meetansh 8

As a faculty member

So that I can add my feedback about the candidate

I want to add a text review to an application

and give them a score from 1 to 5.

Feature: See other reviews for applicant

Meetansh 5

As a faculty member

So that I can know which faculty members have reviewed an application

I want to see the reviews made by other faculty members on that application

Feature: Edit Reviews

Meetansh 6

As a faculty member

So that I can change my review

I should be able to edit any review I have made.

Feature: Label a pdf

Tanay 5

As a faculty member

So that I can jump to the Personal Statement in the PDF

I want to have the table of contents added to the PDF of the application

Feature: Fix default user level

Tanay 5

As a developer

So that I have a user level as faculty for a new account

I want to have a default user level set

Feature: Fix erroneous database migrate

Tanay 6

As a developer

So that I have the correct db schema

I want to have the correct migration files for the models

Feature: Instructions for Deploy and Test

Ruofei 2

As a developer

So that I can deploy and test the projects

I want to follow the README.md and deploy/test the projects

Feature: Move applicants table into database project

Ruofei 5

As a developer

So that I can make operations on applicants table in the database project

I want to move the applicants table into the database management project

Feature: Link File Upload to Applicant Database

Brian 4

As a administrator

So that the PhD Application Review App can have data for users to review

I want to append data from a CSV upload to the Applicant database

Feature: Added Admin Login to Database App

Brian 4

As a developer

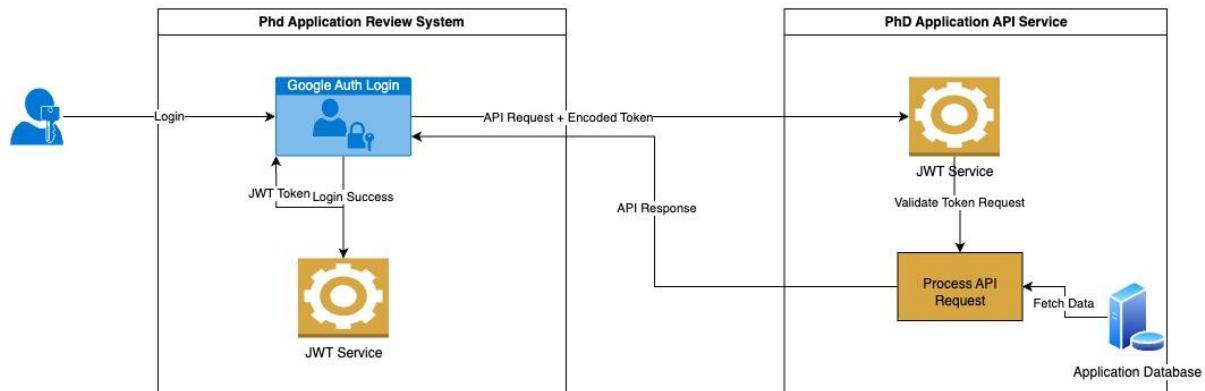
So that only authorized users add data to the Database application

I want to make a login page to authorize the administrator before they make changes

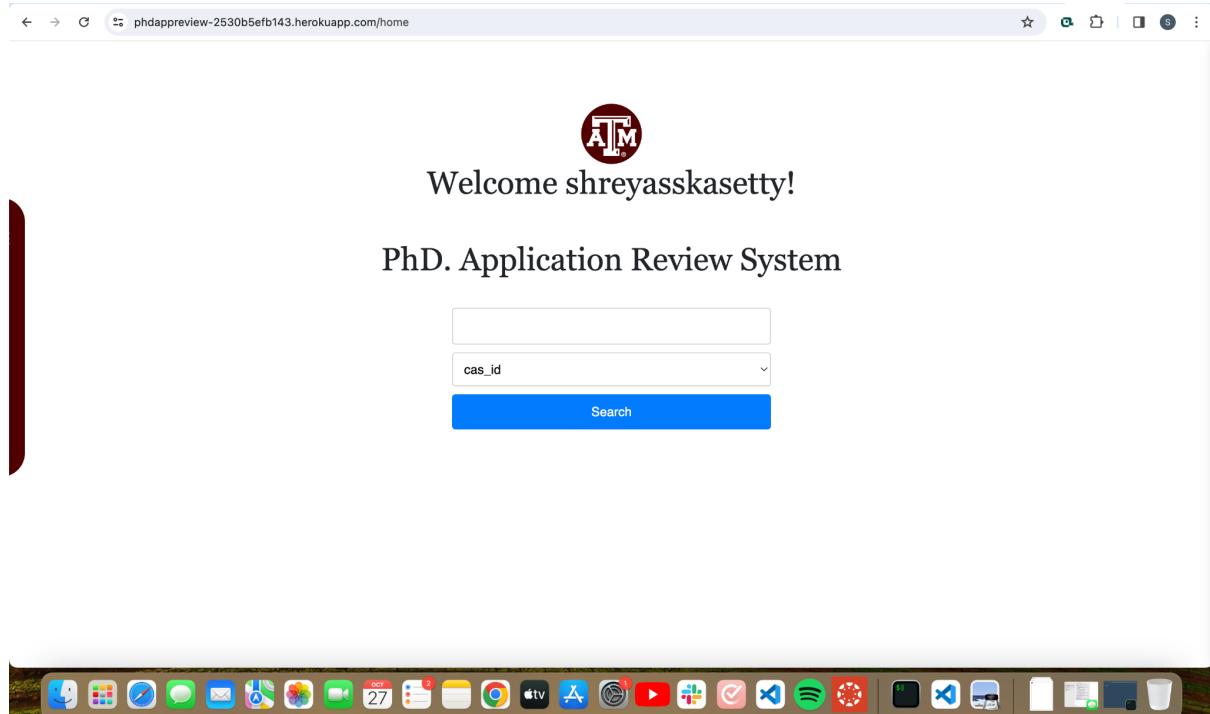
Application Diagram is not changed in this iteration.

Secure API Authentication Design

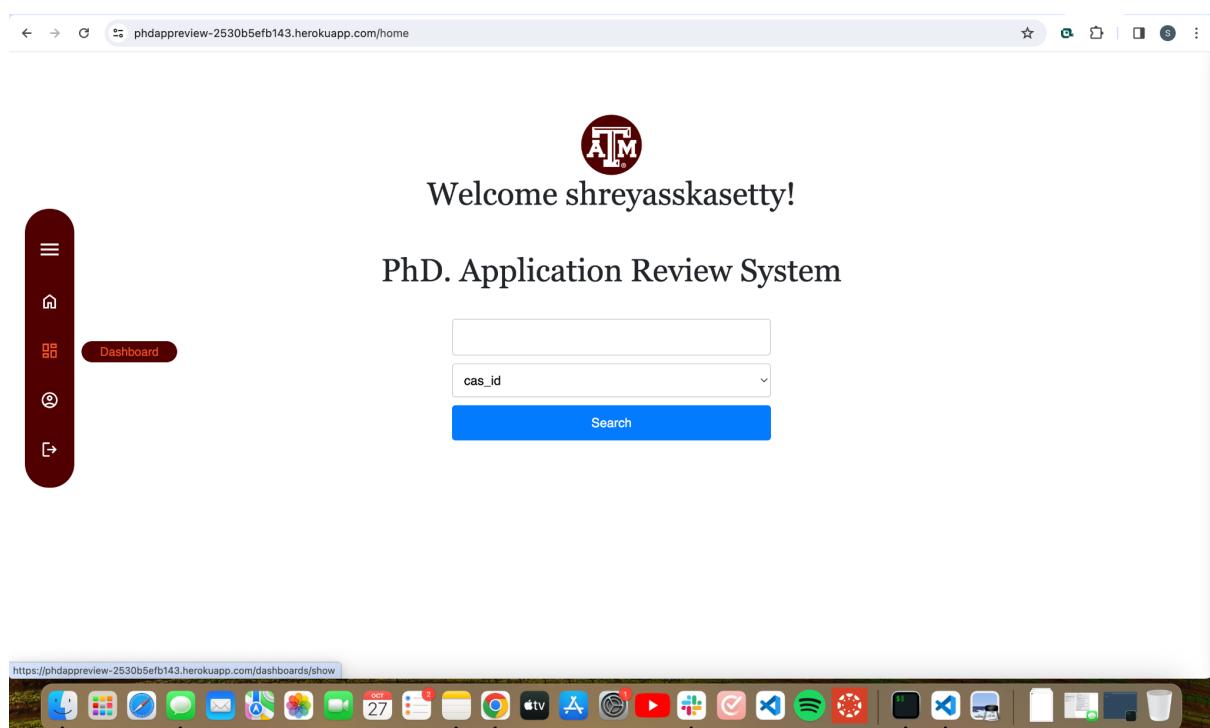
Design of API Authentication for API endpoints in TAMU Application Review Database Service



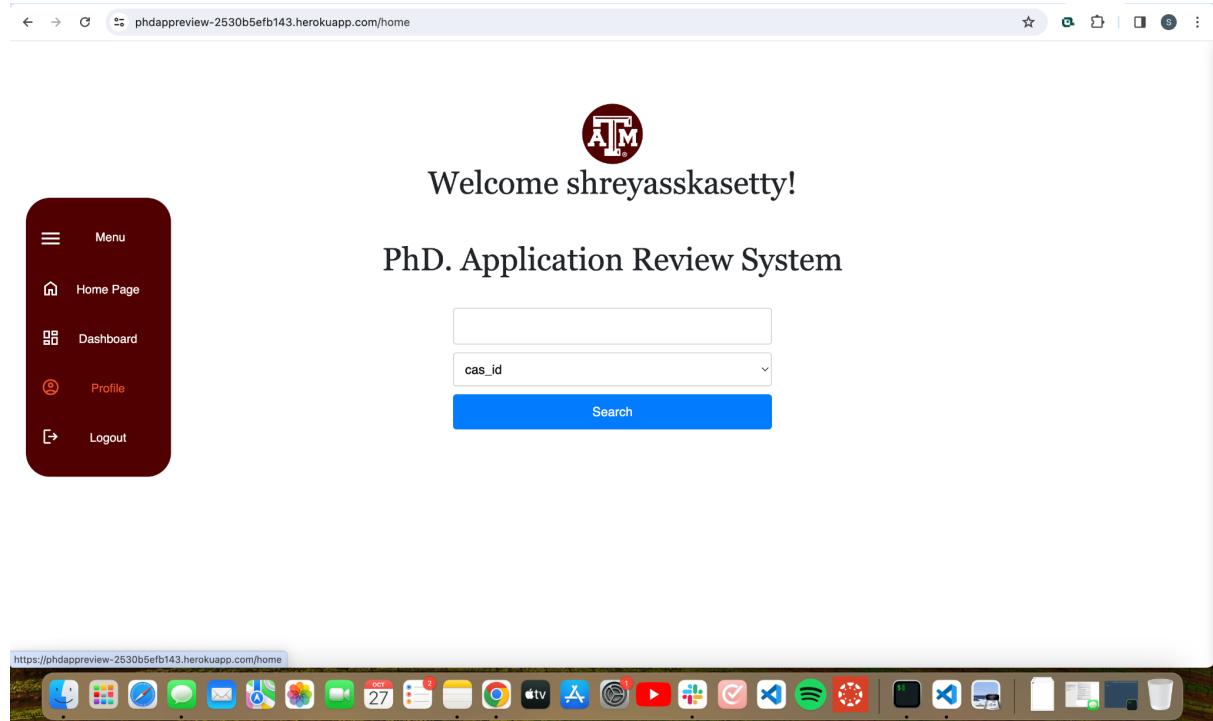
Secure API Design



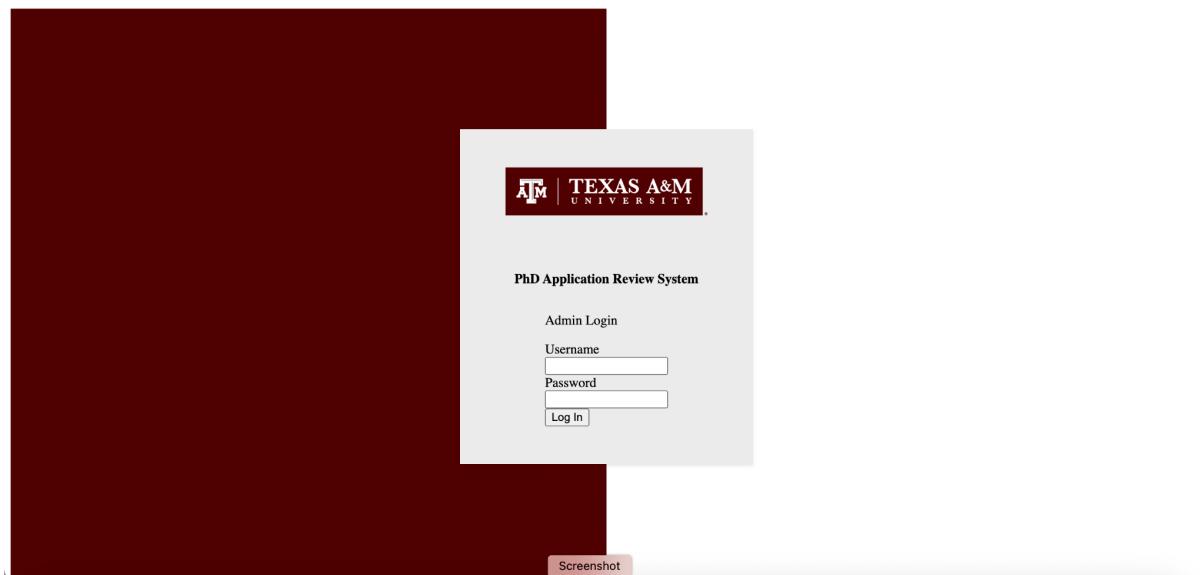
Hidden Nav Bar UI



Unhidden Nav Bar UI



Nav Bar UI Menu Click



Login for Database Application

Database System:

Choose a file to upload: No file chosen

Added Logout Button to File Upload Page



File uploaded successfully

Added Page after successful file upload, with relevant buttons for the user

Test Report:

The screenshot shows a Cucumber Reports test report. At the top, there's a header with the Cucumber Reports logo and a "Log in with GitHub" button. A yellow warning bar says: "⚠ This report will self-destruct in a day. Keep your future reports forever." Below the bar, a summary table provides key metrics: 6 PASSED (green), 2 UNDEFINED (orange), 75% passed (8 executed), 10 hours ago (Last Run), and 4.61 seconds (Duration). The environment section shows "linux" (Penguin icon) and "ruby 3.2.2" (Ruby icon). The cucumber version is "cucumber-ruby 8.0.0" (Cucumber icon). On the right, a red "Delete Report" button is visible. The main content area lists feature files with their status: > 🟠 features/addreview.feature, > ✅ features/login.feature, > 🟠 features/reviews.feature, and > ✅ features/searchapplicant.feature.

© 2023 SmartBear Software. All Rights Reserved.
Privacy | Terms of use



Cucumber Coverage Report

All Files (73.06%)	Controllers (64.67%)	Channels (100.0%)	Models (100.0%)	Mailers (100.0%)	Helpers (100.0%)	Jobs (100.0%)	Libraries (100.0%)	Ungrouped (100.0%)	Generated 4 minutes ago
File	% covered	Lines	Relevant Lines	Lines covered	Lines missed	Avg. Hits / Line			Search: <input type="text"/>
app/controllers/reviews_controller.rb	38.24 %	71	34	13	21	0.76			
app/controllers/tasks_controller.rb	40.54 %	76	37	15	22	0.78			
app/controllers/users_controller.rb	61.11 %	73	36	22	14	1.00			
app/controllers/searches_controller.rb	85.71 %	12	7	6	1	2.00			
app/controllers/sessions_controller.rb	96.00 %	52	25	24	1	2.00			
app/channels/application_cable/channel.rb	100.00 %	4	2	2	0	2.00			
app/channels/application_cable/connection.rb	100.00 %	4	2	2	0	2.00			
app/controllers/application_controller.rb	100.00 %	22	15	15	0	8.87			
app/controllers/dashboards_controller.rb	100.00 %	4	2	2	0	2.00			
app/controllers/homepage_controller.rb	100.00 %	2	1	1	0	2.00			
app/controllers/welcome_controller.rb	100.00 %	14	10	10	0	3.20			
app/helpers/application_helper.rb	100.00 %	2	1	1	0	2.00			
app/helpers/csv_helper.rb	100.00 %	2	1	1	0	2.00			
app/helpers/dashboards_helper.rb	100.00 %	2	1	1	0	2.00			
app/helpers/reviews_helper.rb	100.00 %	2	1	1	0	2.00			
app/helpers/searches_helper.rb	100.00 %	2	1	1	0	2.00			
app/helpers/sessions_helper.rb	100.00 %	2	1	1	0	2.00			
app/helpers/tasks_helper.rb	100.00 %	2	1	1	0	2.00			
app/helpers/users_helper.rb	100.00 %	2	1	1	0	2.00			
app/helpers/welcome_helper.rb	100.00 %	2	1	1	0	2.00			
app/helpers/xlsx_helper.rb	100.00 %	2	1	1	0	2.00			
app/jobs/application_job.rb	100.00 %	7	1	1	0	2.00			
app/mailers/application_mailer.rb	100.00 %	4	3	3	0	2.00			
app/models/applicant.rb	100.00 %	6	5	5	0	2.00			
app/models/application_left.rb	100.00 %	3	2	2	0	2.00			
app/models/application_record.rb	100.00 %	3	2	2	0	2.00			
app/models/gre.rb	100.00 %	3	2	2	0	2.00			
app/models/review.rb	100.00 %	4	1	1	0	2.00			
app/models/school.rb	100.00 %	3	2	2	0	2.00			
app/models/task.rb	100.00 %	16	9	9	0	1.89			
app/models/toefl.rb	100.00 %	3	2	2	0	2.00			
app/models/user.rb	100.00 %	4	3	3	0	2.00			
app/services/search_service.rb	100.00 %	11	6	6	0	2.50			

Showing 1 to 33 of 33 entries

Generated by simplecov v0.22.0 and simplecov-html v0.12.3
using Cucumber Features, RSpec

Rspec Coverage Report

Iteration 4

Feature: Restrict the admin dashboard page

Ruofei 5

As a developer

So that I only allow the users who have the admin role to view the motor admin page

I want to check the user_token before allowing user to get access to the admin page

Feature: JWT authentication for database API

Shreyas, Ruofei 4, 4

As a developer

So that I only allow the users who logged in to use the database APIs

I want to implement the JWT authentication before using APIs for database

Feature: Test coverage

Tanay Patankar 4

As a developer,

In order to ensure the proper working of developed code,

I want to have tests that cover at least 90% of the code.

Feature: Bug Fixes

Tanay Patankar 4

As a developer,

In order to fix issues in the code,

I want to resolve bugs in various components of the system.

Feature: User Access Control

Tanay Patankar 4

As an unauthenticated user,

In order not to access pages when I am not logged in,

I want to disallow my access to these pages and be redirected to the login page.

Feature: Reset Database:

Brian Munoz 6

As an administrator

In order to clear the entire database

I want to have a button that I can click that will clear the database of all entries.

Feature: Rating as radio buttons

Meetansh 2

As a faculty member

So that I can rate the applications from 1 to 5

I want to be able to click on any one of the five radio buttons

Feature: See other reviews by hovering

Meetansh 4

As a faculty member

So that I can see reviews given by other faculty members

I should be able to hover over their names and view the reviews

Feature: Bug fix when adding new reviews

Meetansh 4

As a developer

So that I can ensure there is no issue while adding new reviews

I want to resolve logic errors and bugs in my code

Feature: Admin Dashboard

Shreyas 6

As an admin

So that control items in the database

I should be able to perform basic CRUD operations for tables in my database

Feature: UI Refactor

Shreyas 6

As a user

So that I can easily navigate through the app

I should be able to intuitively interact the menu bar and the system

Feature: Tags

Francesco 6

As a user

so that I can group applications together

I want to be able to add a tag

and only I should be able to view the tag

unless I set it to global

and only I should have full crud access for it

Feature: Database Flexibility:

Francesco 6

As a developer

so that we can enable a flexible system

I want to allow any Excel sheet structure to be supported

and any new fields should be added

and any old fields can be renamed or set to not used

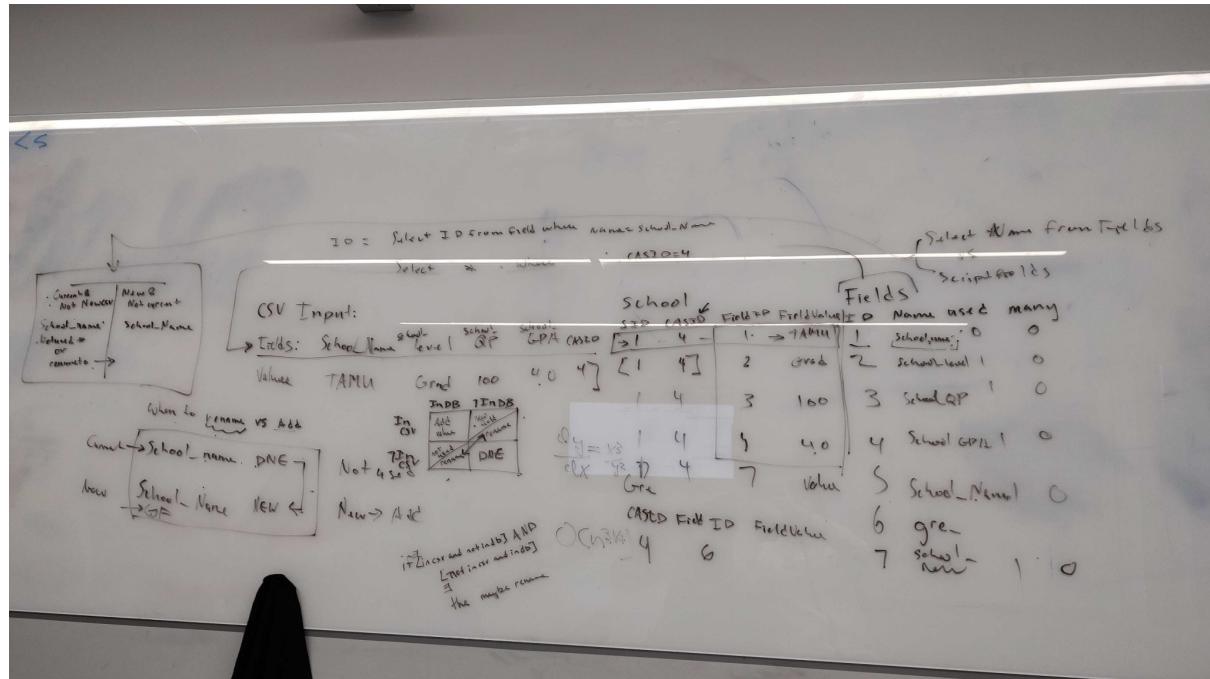
So I designed the database structure

and outlined the logical process we need to code

Application Diagram is not changed in this iteration.

API Secure Design is not changed in this iteration.

The database system is not changed in this iteration.

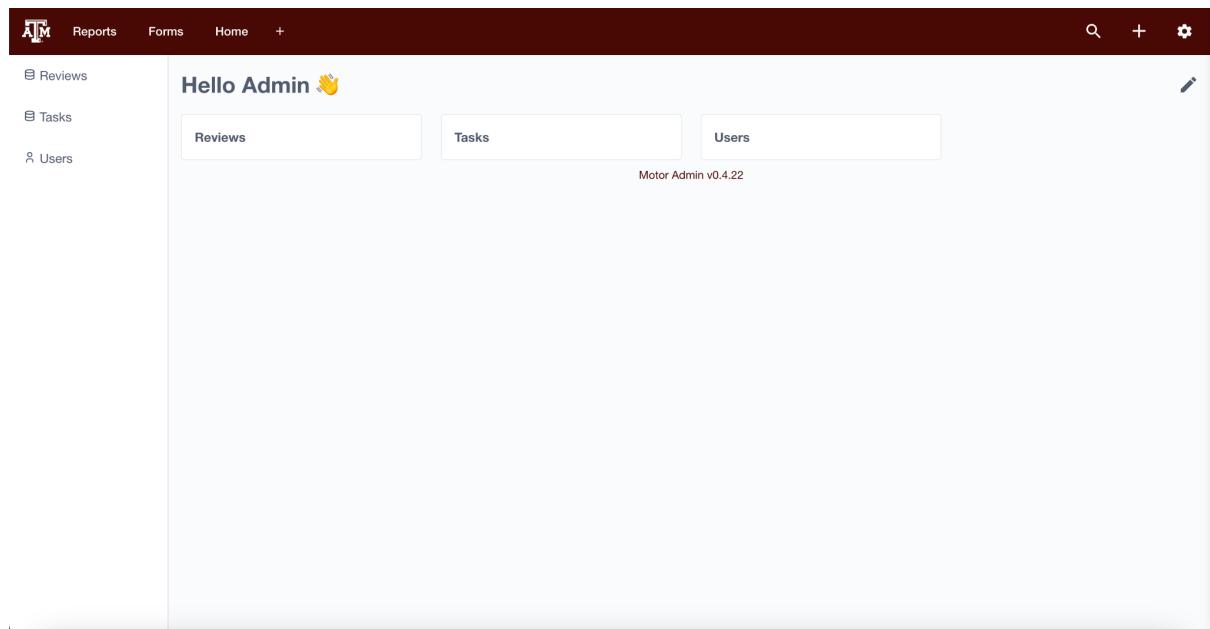


Proposed Database Design

The screenshot shows the homepage of the PhD Application Review System. At the top, there is a dark red header bar with navigation links: "Menu", "Home Page", "Dashboard", "Tasks", "Profile", and "Logout". Below the header is the Texas A&M University logo. A welcome message "Welcome shreyas!" is displayed. The main title "PhD. Application Review System" is centered. Below the title is a search form with three input fields: a large empty input field, a dropdown menu labeled "cas_id", and a blue "Search" button.

The screenshot shows the search results page of the PhD Application Review System. The top navigation bar is identical to the home page. The search form has "India" entered in the first input field and "cas_id" selected in the dropdown. Below the search form, a message "Total Results: 6" is displayed above a table. The table has columns: "ID", "Name", "Email", and "Degree". The results are as follows:

ID	Name	Email	Degree
8914560391	Dewayne, Shreya Bess	cristopher.catherine@yahoo.com	Computer Engineering
6614544379	Kaleigh, Jazlyn Culpepper	natali.michaela@school.edu	Computer Engineering
3034589529	Ramiro, Jaime	winston.devin@protonmail.com	Computer Engineering
3856978770	Jacques, Shreya Otto	cayla.catherine@yahoo.com	Computer Science
9570272315	Prince, Kain Rosenbaum	cristina.kassie@school.edu	Computer Science
12345678	Tanay Patakar	tanay@tamu.edu	Computer Science



Admin Dashboard UI

Reviews	Reviews										
Tasks	ID	Review	User netid	Applicant	Review info	Rating	Status	Gar	Gat	Updated at	Created at
	#4	123	8914560391	The Wok	5	completed	✗	✓		Nov 10, 2023, 11:15 AM	Nov 10, 2023, 1:5
	#3	1234	1123288204	Awesome		completed	✓	✓		Nov 8, 2023, 10:53 AM	Nov 8, 2023, 10:53 A
	#2	shreyasskasetty	1123288204	Not good	4	completed	✓	✗		Nov 7, 2023, 2:46 PM	Nov 7, 2023, 2:46 PM
	#1	123	1123288204	reviewed	2	assigned	✗	✗		Nov 6, 2023, 10:51 AM	Nov 6, 2023, 10:31 A

Admin Database Access UI

My Applications

In Progress Assigned Completed

Applicant	Status	Actions
1123288204	assigned	<button>View</button>
8914560391	completed	<button>View</button>
3034589529	inprogress	<button>View</button>
9570272315	inprogress	<button>View</button>
12345678	completed	<button>View</button>

© 2023 Texas A&M University

Tasks Todo Page

Test Report:

⚠ This report will self-destruct in a day. Keep your future reports forever.

9 PASSED

100% passed

9 executed

an hour ago

Last Run

3.86 seconds

Duration



linux



ruby 3.2.2



cucumber-ruby 8.0.0

trash bin icon Delete Report

- > ✓ features/login.feature
- > ✓ features/reviews.feature
- > ✓ features/searchapplicant.feature

Test Coverage Report

The screenshot shows a test coverage report with the following details:

- All Files (91.4% covered at 3.16 hits/line)**
- 34 files in total.**
- 180 relevant lines, 170 lines covered and 16 lines missed (91.4%).**
- Generated about 7 hours ago.**

File	% covered	Lines	Relevant Lines	Lines covered	Lines missed	Avg. Hits / Line
app/controllers/tasks_controller.rb	93.94 %	20	11	7	4	1.27
app/controllers/users_controller.rb	85.00 %	36	20	17	3	1.25
app/controllers/searches_controller.rb	85.71 %	12	7	6	1	2.57
app/controllers/application_controller.rb	86.67 %	22	15	13	2	12.07
app/controllers/reviews_controller.rb	87.88 %	62	33	29	4	1.88
app/models/task.rb	88.89 %	16	9	8	1	1.78
app/controllers/sessions_controller.rb	96.30 %	58	27	26	1	1.93
app/channels/application_cable/channel.rb	100.00 %	4	2	2	0	2.00
app/channels/application_cable/connection.rb	100.00 %	4	2	2	0	2.00
app/controllers/dashboard_controller.rb	100.00 %	4	2	2	0	2.00
app/controllers/homepage_controller.rb	100.00 %	2	1	1	0	2.00
app/controllers/welcome_controller.rb	100.00 %	19	12	12	0	4.67
app/helpers/application_helper.rb	100.00 %	2	1	1	0	2.00
app/helpers/rails_vv_helper.rb	100.00 %	2	1	1	0	2.00
app/helpers/dashboards_helper.rb	100.00 %	2	1	1	0	2.00
app/helpers/reviews_helper.rb	100.00 %	2	1	1	0	2.00
app/helpers/searches_helper.rb	100.00 %	2	1	1	0	2.00
app/helpers/sessions_helper.rb	100.00 %	2	1	1	0	2.00
app/helpers/tasks_helper.rb	100.00 %	2	1	1	0	2.00
app/helpers/users_helper.rb	100.00 %	2	1	1	0	2.00
app/helpers/welcome_helper.rb	100.00 %	2	1	1	0	2.00
app/helpers/xlsx_helper.rb	100.00 %	2	1	1	0	2.00

Test Coverage Report

Iteration 5

Feature: Add Dynamic User Interface for Database:

Brian Munoz 14

As an administrator

In order to dynamically handle elements from a spreadsheet and database

I want to have a user interface to rename, discard, or add fields

Feature: Refactor JS and CSS code:

Brian Munoz 2

As a developer

In order to make our codebase more readable and organized

I want to remove JS and CSS code from html.erb files

Feature: Add Routes and Controller Methods to Handle GUI input:

Brian Munoz 4

As a developer

In order to make sure user input gets passed from the Database GUI to the controller methods

I want to add POST routes and methods to handle input

Feature: Database Implementation and refactoring:

Francesco 3

As a developer

so that we can enable a flexible system

I want to refactor the database

And add flexible upload logic

And update the search function

And update where all hard-coded use cases from the old database

Feature: Reviews functionality on the front-end

Meetansh 5

As a user

So that I can add reviews to applications

I should be able to write text

And rate the application on a scale of 1-5

And recommend the applicant for a GAT/GAR

Feature: Renew expired JWT token

Ruofei 5

As a user

So I should know it when the JWT token is expired

I can log in again and get a new token

Feature: Add cucumber tests for task and user controllers

Ruofei 5

As a developer

So I should run tests for all controllers

I want to improve the coverage rate for the task and user controllers

Feature: Add tables for refactored database

Ruofei 4

As a developer

So I can have the tables that have new fields

I want to have tables for the refactored database

Feature: Randomly Assign all the Reviews

Tanay 4

As a committee chair

So I can assign all the applications to reviewers in one click

I want to have a button that I can click that will assign all the applications

Feature: Dashboard Page for Committee Chair

Shreyas 2

As a user with committee_chair user role

I want to be able to open Dashboard Page

So that I can assign applications to the committee members

Feature: Application Review Page:

Shreyas 2

As a user

I want to be able to open an application

So that I can view the application information and add reviews

Application Diagram is not changed in this iteration.

API Secure Design is not changed in this iteration.

The screenshot shows the homepage of the PhD Application Review System. At the top, there is a dark navigation bar with links for 'Menu', 'Home Page', 'Dashboard', 'Tasks', 'Profile', and 'Logout'. Below the navigation bar, the Texas A&M logo is displayed, followed by the welcome message 'Welcome shreyas!'. The main title 'PhD. Application Review System' is centered above a search form. The search form consists of two input fields: the first is empty, and the second is a dropdown menu labeled 'cas_id'. Below the dropdown is a blue 'Search' button. At the bottom of the page, a dark footer bar contains the text '© 2023 Texas A&M University'.

Home Page UI

The screenshot shows a search interface with two input fields: one for 'India' and another for 'cas_id'. A blue 'Search' button is below them. Below the search bar, a message says 'Total Results: 6'. A table follows, with columns: ID, Name, Email, and Degree. The data is as follows:

ID	Name	Email	Degree
8914560391	Dewayne, Shreya Bess	cristopher.catherine@yahoo.com	Computer Engineering
6614544379	Kaleigh, Jazlyn Culpepper	natali.michaela@school.edu	Computer Engineering
3034589529	Ramiro, Jaime	winston.devin@protonmail.com	Computer Engineering
3856978770	Jacques, Shreya Otto	cayla.catherine@yahoo.com	Computer Science
9570272315	Prince, Kain Rosenbaum	cristina.kassie@school.edu	Computer Science
12345678	Tanay Patakar	tanay@tamu.edu	Computer Science

Search Results UI

The screenshot shows a section titled 'Committee Members' with a heading 'Shreyas Shivakumar'. Below it is a table with columns: Name, CAS ID, and Actions. Two rows are listed, each with a 'remove' button in the Actions column.

Name	CAS ID	Actions
Default Name	2	<button>remove</button>
Default Name	5	<button>remove</button>

© 2023 Texas A&M University

Committee Chair Dashboard UI

Home Dashboard Tasks Logout

Brant, Antonio Patel

Field	Value
CAS ID	1123288204
Name	Brant, Antonio Patel
Gender	Male
Citizenship Country	United States
Date of Birth	N/A
Email	prince.tanner@school.edu
Degree	Computer Science
Application Submitted	Apr 12, 3456 at 6:04 AM
Status	Complete
Research Interests	research_interests11111

Add your review here

Other Reviews

Application Review Page

Menu Home Page Dashboard Tasks Profile Logout

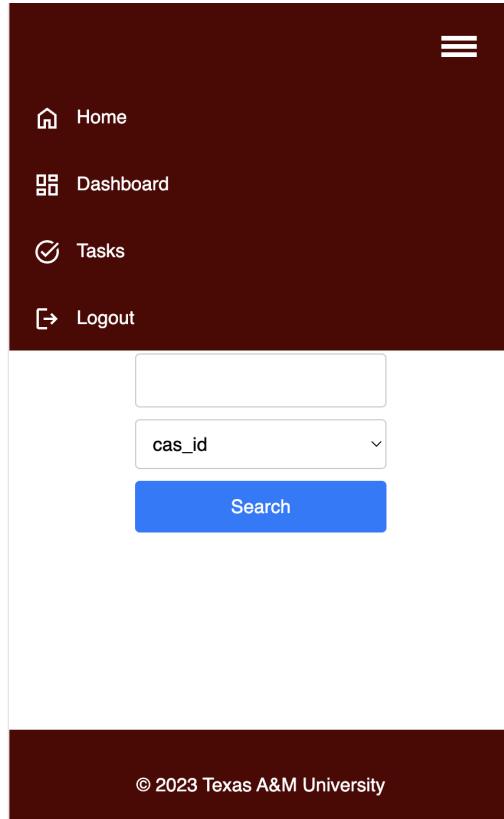
My Applications

In Progress Assigned Completed

Applicant	Status	Actions
1123288204	assigned	<button>View</button>
8914560391	completed	<button>View</button>
3034589529	inprogress	<button>View</button>
9570272315	inprogress	<button>View</button>
12345678	completed	<button>View</button>

© 2023 Texas A&M University

Application Review Page



Responsive Navbar UI for Mobile View Port

Database System:

Please Handle The Following Elements:

Old Fields (Not in upload, but in the database):

- phone

New Fields (In upload, but not in the database):

- Favorite Name
- birthday

New GUI for new Database Implementation

Test Report:

This report will self-destruct in a day. Keep your future reports forever.

9 PASSED

100% passed
9 executed

an hour ago
Last Run

3.86 seconds
Duration

linux (Penguin icon)

ruby 3.2.2 (Ruby icon)

cucumber-ruby 8.0.0 (Cucumber icon)

Delete Report

> ✓ features/login.feature
> ✓ features/reviews.feature
> ✓ features/searchapplicant.feature

Test Coverage Report

All Files (90.74%) covered at 2.86 hits/line)

Generated 5 days ago

All Files (90.74%) covered at 2.86 hits/line)

35 files in total.
216 relevant lines, 196 lines covered and 20 lines missed. (90.74%)

Search:

File	% covered	Lines	Relevant Lines	Lines covered	Lines missed	Avg. Hits / Line
app/controllers/applicant_controller.rb	71.43 %	19	7	5	2	1.00
app/controllers/tasks_controller.rb	72.22 %	36	18	13	5	1.44
app/controllers/application_controller.rb	85.00 %	30	20	17	3	9.90
app/controllers/users_controller.rb	85.71 %	36	21	18	3	1.29
app/controllers/searches_controller.rb	88.89 %	13	9	8	1	2.11
app/models/task.rb	88.89 %	16	9	8	1	1.78
app/controllers/reviews_controller.rb	90.24 %	82	41	37	4	1.80
app/controllers/sessions_controller.rb	96.30 %	58	27	26	1	1.93
app/channels/application_cable/channel.rb	100.00 %	4	2	2	0	2.00
app/channels/application_cable/connection.rb	100.00 %	4	2	2	0	2.00
app/controllers/dashboards_controller.rb	100.00 %	4	2	2	0	2.00
app/controllers/homepage_controller.rb	100.00 %	2	1	1	0	2.00
app/controllers/welcome_controller.rb	100.00 %	19	12	12	0	5.00

Test Coverage Report

Links:

Deployed Application:

<https://phdappreview-2530b5efb143.herokuapp.com/>

We have split the database with FERPA data in a separate rails API hosted at
<https://phdapplicationsystemdatabase-5e5fc255409c.herokuapp.com>

GitHub Repo:

<https://github.com/RubyOffTheRailsTAMU/PhD-Application-Review-System>

Project Management Page:

<https://github.com/orgs/RubyOffTheRailsTAMU/projects/1>

Slack Channel:

Phdapplicatio-buc1202.slack.com

Presentation Video:

<https://youtu.be/KGvXCwMYCAA>

Project Demo Video:

https://youtu.be/8Q_FJryO32g