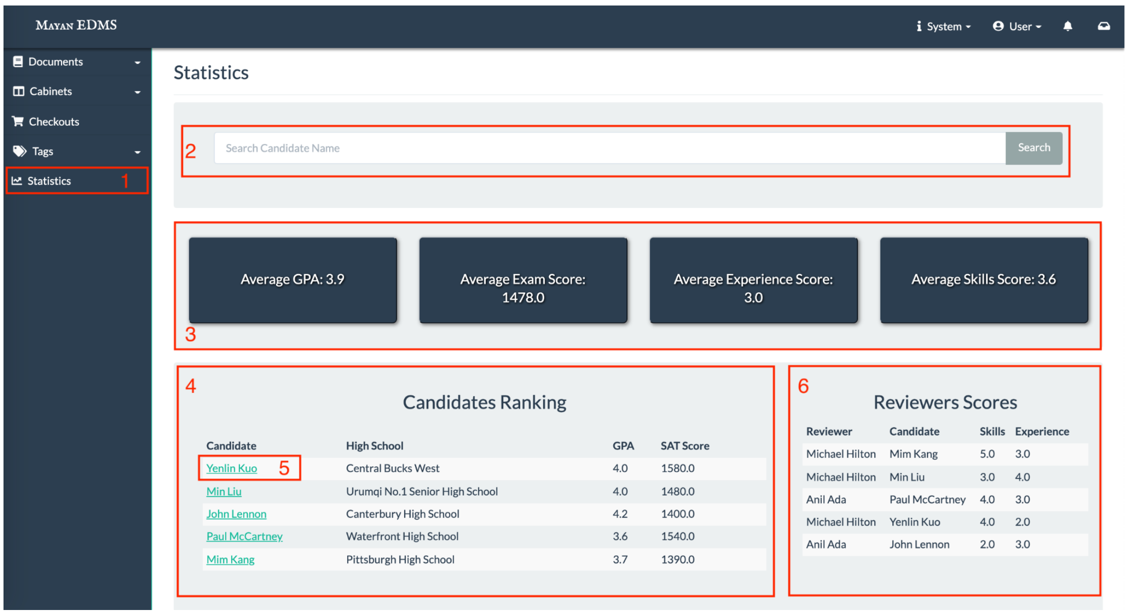
Feature description:

The additional feature implemented a dashboard displaying the candidates and aggregated statistics like average GPA/Exam score per candidate; average review score like Experience/Skill score per candidate, and also includes all the candidates ranking and review score in a sheet format about the candidate pool.

Using the Dashboard



Average GPA: 3.9

Average Exam Score: 1478.0

Average Experience Score: 3.0

Average Skills Score: 3.6

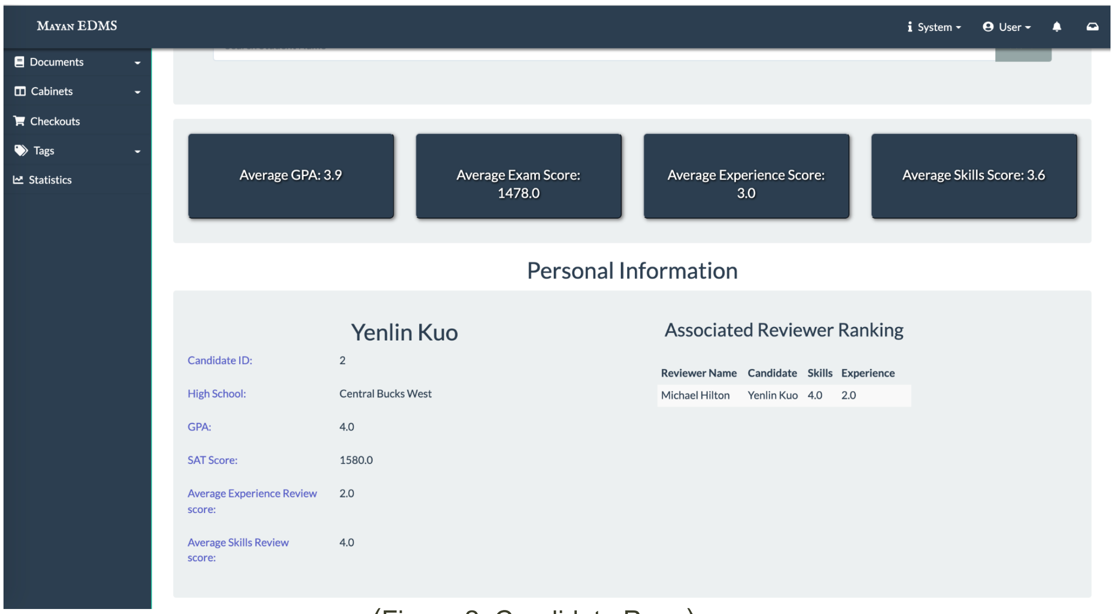
Candidates Ranking

Candidate	High School	GPA	SAT Score
Yenlin Kuo	Central Bucks West	4.0	1580.0
Min Liu	Urumqi No.1 Senior High School	4.0	1480.0
John Lennon	Canterbury High School	4.2	1400.0
Paul McCartney	Waterfront High School	3.6	1540.0
Mim Kang	Pittsburgh High School	3.7	1390.0

Reviewers Scores

Reviewer	Candidate	Skills	Experience
Michael Hilton	Mim Kang	5.0	3.0
Michael Hilton	Min Liu	3.0	4.0
Anil Ada	Paul McCartney	4.0	3.0
Michael Hilton	Yenlin Kuo	4.0	2.0
Anil Ada	John Lennon	2.0	3.0

(Figure 1. Statistics Page)



Average GPA: 3.9

Average Exam Score: 1478.0

Average Experience Score: 3.0

Average Skills Score: 3.6

Personal Information

Yenlin Kuo

Candidate ID:	2
High School:	Central Bucks West
GPA:	4.0
SAT Score:	1580.0
Average Experience Review score:	2.0
Average Skills Review score:	4.0

Associated Reviewer Ranking

Reviewer Name	Candidate	Skills	Experience
Michael Hilton	Yenlin Kuo	4.0	2.0

(Figure 2. Candidate Page)

1. The dashboard can be found under the **statistics** tab on the navigation bar. (Figure 1)

2. The search bar allows users to look up specific candidates in the database by name. This feature is currently in progress, so searching is currently unavailable. (Figure 1)
3. The top section displays averages of SAT scores and GPA aggregated across all candidates. Additionally, the average skills score and experience score of all candidates given by reviewers are also shown. This feature gives the user an overview of the candidate pool. (Figure 1)
4. The next section organizes a view for all candidates based on a ranking that combines their SAT out of 1600 and GPA out of 4.0. (Figure 1)
5. The name is linked to the candidate's page with their information. To access the candidate's information, click on the name. The site will direct to the candidate's record including their personal information and reviews related to the particular candidate.(Figure 2)
6. The reviewer's scores section organizes a view for the reviewers and their reviews based on the combined score for both skills and experiences. (Figure 1)

Improving the Dashboard

Developers can improve on the dashboard and add their own features.

- To modify the statistics page and students page, developers can modify the `statistics.html` and `students.html` templates within the `/appearances/templates/appearance` directory.
- The views for the dashboard are located with the 'common' directory, whereas the models for the dashboard are located within the` /display/` directory which also contains migrations that create tables and student/reviewer entries within '/migrations/' as well as tests within '/tests/' for quality assurance on the loadability of views.
- Building the docker container simply involves following the steps for building noted in the official Mayan documentation.
- However, we do note that at times Docker may throw a "Redis Lock" error which requires developers to rerun the "docker-compose" command until the error is resolved. (Failure to do so may result in localhost not loading)

Testing the Feature

We did manual testing through the following:

- By running through the web pages through localhost and determining if the site was accessible and page content appeared where it should be.
- We checked migrations and insertions to the database by seeing if the sql commands actually ran in the dbshell through the docker container and if the tables were updated both on the database and on the frontend when loading candidates/reviews.
- Navigation of the site was checked via clicking through associated links and making sure that all links were not broken (i.e. linking from a particular candidate on the statistics page to their specific profile); through this process, we discovered that the search bar (currently WIP) had broken navigation.
- We also ran a Lighthouse report on both pages to test that our project met industry standards. Our scores are shown below.

  Statistics Page:

Performance	Accessibility	Best Practices	SEO
72	90	100	78

  Candidates Page:

Performance	Accessibility	Best Practices	SEO
71	90	100	78

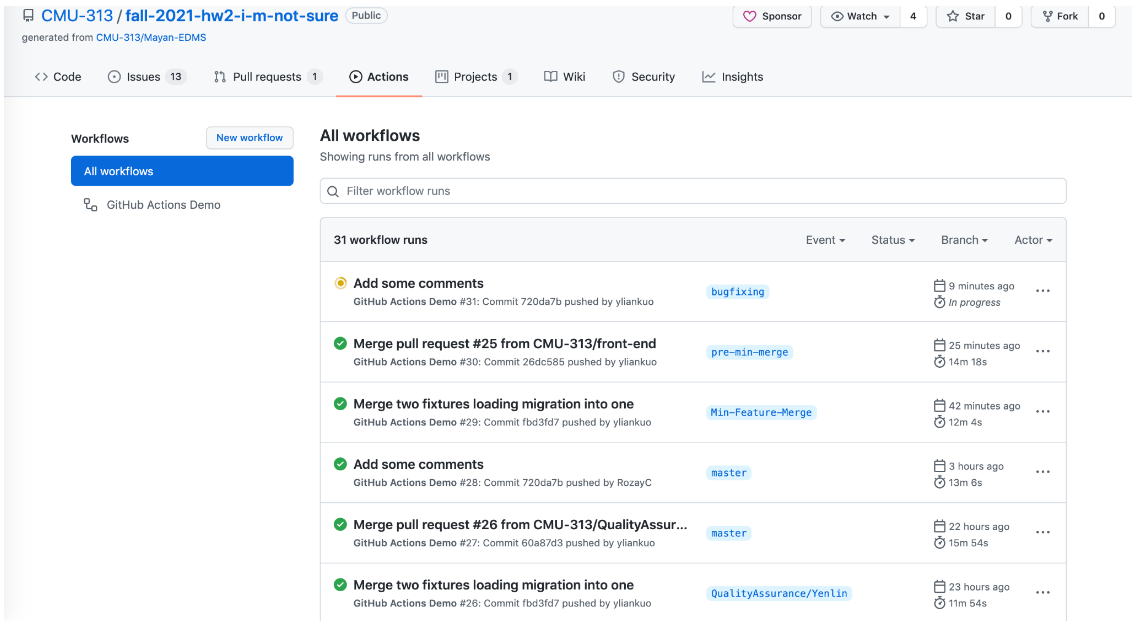
  As a work in progress, we intended to improve on the Lighthouse scores, however, we held back on doing so since many of the issues were those related to the general Mayan site rather than the design of our individual websites.

We also developed automated tests for site accessibility in terms of checking that certain views were accessible (Status 200) which we placed within the '/tests' directory for continuous integration testing.

We built the Continuous Integration (CI) pipeline for testing our code.

- The CI is run on GitHub Actions, where the pipeline is triggered by each PR commit or commits on the master branch.
- For each CI build, we first build Mayan through `docker-compose build` to make sure that our changes do not disrupt the build process for Mayan EDMS and cause it to break.

- Then we run `make docker-runtest-all` and our custom test script under apps/display to make sure the new and existing functionalities work as expected.
- As the CI pipeline was causing issues with docker-compose and the loading of localhost, we've currently omitted it from the master branch (due to issues with compilation and docker-compose missing files within the 'scantree'/'whitenoise' static folders). Instead, to test using the automated CI pipeline, pushes were made to the bug fixing branch.
- An example of our CI working properly on the bugfixing branch (Figure 3)



All workflows
Showing runs from all workflows

Filter workflow runs

31 workflow runs — Event, Status, Branch, Actor

Run	Branch	Time
Add some comments GitHub Actions Demo #31: Commit 720da7b pushed by yliankuo	bugfixing	9 minutes ago In progress
Merge pull request #25 from CMU-313/front-end GitHub Actions Demo #30: Commit 26dc585 pushed by yliankuo	pre-min-merge	25 minutes ago 14m 18s
Merge two fixtures loading migration into one GitHub Actions Demo #29: Commit fbd3fd7 pushed by yliankuo	Min-Feature-Merge	42 minutes ago 12m 4s
Add some comments GitHub Actions Demo #28: Commit 720da7b pushed by RozayC	master	3 hours ago 13m 6s
Merge pull request #26 from CMU-313/QualityAssur... GitHub Actions Demo #27: Commit 60a87d3 pushed by yliankuo	master	22 hours ago 15m 54s
Merge two fixtures loading migration into one GitHub Actions Demo #26: Commit fbd3fd7 pushed by yliankuo	QualityAssurance/Yenlin	23 hours ago 11m 54s

(Figure 3. CI Test)