

Future Backlog

Gabrielle Martone, Joseph Cenci, & Julianna Puleo

*= Front end and back end edits

**= Front end edits only

***= Back end edits only

1. * Single Sign on for Lafayette Email
 - Adding this to the MainPage.java instead of the view with the background picture and 3 buttons that say Student, Alumni, and Employee
 - Try to get ITS to allow for alumni to use the lafayette community sign on
2. ** Employee should be able to edit the project proposals in the project view dialog in ProjectListEmp.java
 - On the front end, add an edit button to the view project pop up window in the ProjectListEmp.java file.
 - Add database functionality from the SQLProjectEmp object in that file to edit that project's attributes in the database.
3. ** Have the project proposal page for Employee have the functionality to parse a document to enter the fields in ProjectProposalEmp.java
 - Instead of an employee copying and pasting the project fields over, the employee should be able to import a document. This document would be the one that gets sent to the email if that capability is added. This would make the employees process more efficient and also would make the parsing easier if the documents are all structured the same from exporting.
4. *** Enhance email functionality
 - All of the email functionality is handled in the EmailSender.java file.
 - Currently, the email functionality prints out each of the project description elements into the body of an email, and sends this email to a Dyer Center Employee. This allows the Dyer Center Employee to view all of the required information, and then copy and paste values into their project proposal page.
 - This could be enhanced by putting all of the project description elements into a document, and then sending that document as an attachment in an email to the Dyer Center Employee. This would allow them to edit the document, and then potentially upload that project document to create a one rather than copy and paste values in.
 - Edit the EmailSender.java file so that recipients of emails can be specified. Since email functionality is still being developed, all of the emails are being sent to

dummydyer@gmail.com, (password: dummypassword). This was to allow testing and checking of the email functionality. In the future, emails will be sent to the actual Dyer Center Employees, and to Students.

- Students should receive emails reminding them about upcoming events a week beforehand.
 - When a student matches with a project, that student should receive an email telling them that they've matched with a new project, and that they should log onto the Dyer Center Web Application to find out more about it.
 - Create an option for Dyer Center Employees to export projects from their Projects page. This would then send the employee the project description file mentioned earlier, that they could then potentially edit, and then re-upload.
5. *** Improve matching algorithm
- Have the matches be on more than just matching on one skill or category.
 - Have the matches be filtered based upon whether any come up based upon skills alone, and if not filter to category for a worst case match.
 - i. This is explained further in the Developer Manual
 - Store in database how many items were matched and which ones matched the student.
6. *** Edit the table Table_Projects_Proposed.
- Have an additional column for status of the project once it is proposed.
7. *** Possible Change → Load from the School Calendar under Dyer Center Tag.
- As of now the calendar is a standalone component.
 - Might be altered to not add events there but rather just load what events have been tagged for the Dyer Center from the Lafayette College Calendar.
 - This would require a lot of communication with IT on discovering how to go about linking the two systems.
 - Use Calendario 2019 project as reference.
8. ** Color code which skills a student matched in the skills list shown in the project view for students in ProjectListStud.java.
9. ** Have the matches projects list show up for in the project list view for the Student initially instead of the full project list loading, ProjectListStud.java.
10. * Add Google analytics to the web page.
- How many users are using on a daily basis?
 - Rate of project proposals?
 - Amount of students signed up, etc.
11. * Have skillset dependent upon major.
- Currently, our skill set is based on a category and then those categories have skill sets.
 - When a student sets up their profile and enters major and minor there should be

an auto population of skills. The student then can unselect the ones they do not have or want.

12. ** Need a default page loaded when each user enters their own page.
 - There is currently a pop up that tells the user to click one of the navigation items to start. With the UI using routing to display the different views, we could not figure out how to initially load a page. May need a different way of changing views.
13. ** Need a hierarchical structure for categories and skills, so they can be collapsed into one line.
 - Add the TreeGrid component from Vaadin, <https://vaadin.com/components/vaadin-treegrid> The tree grid should have a checkbox to be able to select and deselect.
 - Add this to the ProjectListStud.java. When a student views a project, the skill set should be displayed in a TreeGrid instead of a Grid. As well, when a student proposes a new project, the way the user selects the skills should be in a TreeGrid and now a Grid.
 - Add this to the ProjectListEmp.java & ProjectProposalEmp.java & SkillsEmp.java. For the way the skills are displayed in both views
 - Add this to the ProjectListAlum.java & ProjectProposalAlum.java. For the way the skills are displayed in both views.
14. ** Need a auto completion for adding category names & skill name in the SkillsEmp.java page.
 - The field should autocomplete to keep consistency with entering the categories for skills.
 - Autocompletion should be for skills incase the user wants to delete it, they can find it easier.
15. *Automatic filling of the calendars on the event views for Students and Employees.
 - Instead of having the Dyer Center Employees manually input all of their events, the calendars should import all of the Dyer Centers events from Lafayette's Calendar of Events.
16. ** Vaadin 13 is not responsive with its components.
 - The Dyer Center wants the app to be responsive as they feel most students will be using their cell phones to use this web application.
17. * Enhance profiles for Student and Alumni to show the projects proposed and the status of those projects.
 - When each user submits a project proposal, for student → ProjectListStud.java and for alumni → ProjectProposalAlum.java, they should have a way of seeing all the projects they proposed. As well, the status of the proposal either pending, under review, approved, or waiting for a team to be assembled, etc.

- This would track a project from the initial proposal through the completion.