BoilerTrack

CS 30700
Sprint 2 Planning Document

Team 19

- Anitej Waghray
- Daivik Ghosh
- Daniel Benes-Magana
- Mihika Sharma
- Shlok Bairagi
- Shreya Laxmi Nagendra

Sprint Overview:

In this sprint, we aim to complete functionalities that were not fully addressed in the previous sprint, including staff login, password reset, and account deletion features. Additionally, this sprint will focus on implementing new features to streamline the claim process, such as notifications for matched items and a more refined method for users to submit and modify claim requests. We will also begin developing aspects of the dispute process. These improvements aim to enhance the overall usability and functionality of BoilerTrack, ensuring a more robust platform for both students and staff.

Scrum Master: Shlok

Meeting Plan:

- Monday 3:30 PM with Project Coordinator
- Monday 8:00 PM
- Thursday 6:30 PM
- Saturday 5:00 PM

Risks and Challenges:

One of the main challenges in this sprint is ensuring accurate and efficient retrieval of data from the database, such as claim request details, while maintaining optimal performance. Another complex task will be to implement a user-friendly and secure process for claim disputes, especially when dealing with sensitive user data. Additionally, ensuring proper security measures for account deactivation and password resetting is an important task that could be quite challenging. A significant challenge will be data processing and API efficiency for sending notifications to users. It would be risky to not test these functionalities thoroughly. In order to prevent issues during deployment, we must test each component and functionality of our project thoroughly.

Current Sprint Details:

Student-side:

User Story #1

As a student user, I want to have a reset password option.

#	Description	Estimated Time	Owner
1	Add verification for if account exists on the backend and send confirmation or error message	2h	Daniel
2	Add password reset token generation and store it in the database	2h	Daniel
3	Create page to input new password	2h	Daniel
4	Add verification of token and update password in the database	3h	Daniel
5	Create unit test to check correct functionality of password reset by checking database for relevant updates	1h	Daniel

Acceptance Criteria:

- Given the UI for the reset password page has been implemented correctly, the user should be able to navigate to the page and enter their email to be verified
- If the user's account does not exist, an alert should show that it doesn’t exist.
- If the user’s account does exist, the user should receive an email to get their password reset token
- Given the UI for the new password page is implemented correctly, the user should be able to enter a new password and redirect back to the login page
- When the new password is entered, the user should not be able to log in with the old one

User Story #2

As a student user, I want to be able to deactivate my account.

#	Description	Estimated Time	Owner
1	Create a page that gives the option to close an account	2h	Daniel
2	Add confirmation prompts in the form of "Please type <email/username> to confirm you want to delete your account"	2h	Daniel
3	Add final password verification if a user wants to delete their account	1h	Daniel
4	Server should verify that the logged in user is the account that will be removed	1h	Daniel
5	Server should remove the account from the database by marking it as deleted	2h	Daniel
6	Job should be run to remove accounts marked as deleted in bulk during low traffic hours	3h	Daniel

Acceptance Criteria:

- Given the homepage has an option to manage the user account, there should be an option for deleting the user account
- Given that the UI for deactivating an account is set up properly, when the user navigates to this button in the user profile page, they should be presented with dialog boxes to confirm that they want to deactivate their account.
- Given the user decided to deactivate their account, the client should password verify the user before deleting the account
- Given the user information is marked as deleted, the server should remove the data from the database in a regular job
- Given the user confirms they want to delete their account, they should be redirected to the login page after

User Story #3

As a user, I want to view the current status of my flagged items (claimed, in dispute, etc.).

#	Description	Estimated Time	Owner
1	Add a “view claim requests” button to the student user’s profile page	30 mins	Shreya
2	Create UI for student user to see items for which they have submitted a claim request	3 hr	Shreya
3	Fetch user’s claim requests details from the database	1 hr	Shreya
4	Create backend API endpoints to return details about user’s claim requests from the database	3 hr	Shreya
5	Create unit tests to test correct functionality of user viewing claim requests	2 hr	Shreya

Acceptance Criteria:

- The ‘view claim requests’ button should be visible and accessible on the user’s profile page.
- When a user clicks the ‘view claim requests’ button, they should be redirected to a page displaying a list of their claim requests.
- The UI for viewing claim requests should clearly display each item for which a claim request has been submitted, including relevant details such as item name, description, and status (e.g: claimed, in dispute, etc).
- The backend should correctly retrieve and return the claim requests details from the database
- The backend should retrieve and return details for items for which only the logged-in user has submitted a claim request.

User Story #4

As a user, I want to pre-input details like description, date, and location about my lost item to help match found items.

#	Description	Estimated Time	Owner
1	Create UI form for users to be able to enter item details	2 hr	Mihika
2	Connect/send item data to be stored in database via backend routing	3 hr	Mihika
3	Create edit,delete option to modify form details	3 hr	Mihika
4	Add lost item request to other requests associated to user in db	2 hr	Mihika
5	Create ui to view all lost item requests	2 hr	Mihika

Acceptance Criteria:

- Given that the UI for inputting lost item details is available to student-user, student user should be able to input relevant details and be prompted an error if a required field is left blank.
- Given that the student-user was able to input and submit lost-item-request details, this request should be stored in the respective database and be associated with the student-user id.
- The student-user should be able to view a list of their lost-item-requests and their current status.
- The student-user should be able to edit a lost-item-request and have the requests listing updated subsequently.
- The student should be able to delete a lost-item request and have the requests listing updated subsequently.

User Story #5

As a user, I want to receive a notification when an item matching my description of a lost item is found.

#	Description	Estimated Time	Owner
1	Create UI for notification tab for student-user that contains all user-specific notifications	2 hr	Mihika
2	Update status of request in lost-item-requests listing (pending, match-found, claimed, etc)	3 hr	Mihika
3	Create button for user to go to item view of item that matches description	2 hr	Mihika
4	Update lost-item-request status in database (pending, match-found, claimed, etc)	2 hr	Mihika
5	Create backend functionality to check for “item matches” between items in the database and lost-item-requests logged by user	3 hr	Mihika

Acceptance Criteria:

- Given that a lost- item-request has been entered by a student-user, a user should get a notification within 24 hours of an unclaimed item being entered into the system.
- Student-user should be able to view all such notifications on clicking the notification tab on their dashboard
- The lost-item-requests listing for student-user should be updated such that the request for which a match is found displays an updated status.
- Given that a match is identified, the user should be able to navigate to item view of the matched item through their lost-items-requests listing.
- Given that a student claims an item listed in their lost-item-requests, the user should be able to mark the request as “completed” within their lost-item-requests listing.

User Story #6

As a user, I want to claim my item through a verification process (e.g., providing additional details or proof).

#	Description	Estimated Time	Owner
1	Add “Submit Claim Request” button to the item detail page and a “Submit Proof” UI form	1 hr	Shlok
2	Accept all relevant and valid data from the user	1 hr	Shlok
3	Submit the data to the backend to store for review	2 hr	Shlok
4	Send a notification to the staff location where the item was found about the claim request	5 hr	Shlok

Acceptance Criteria:

- If a student feels that an item is theirs, when they are on the individual item view, then they should have an option to submit a claim request for their item.
- Given that the students press the claim request button, then a form should open up that asks for more details.
- Given the student fill in the details, they should be validated to check if they pass basic criteria
- Given that a claim request was successfully added to the database, a notification should be sent to the staff at the found location, about a new claim request.

User Story #7

As a user, I want to modify my claim on a lost item in case of error.

#	Description	Estimated Time	Owner
1	The student should be able to view all their past claim requests	1 hr	Shlok
2	Add a modify request button to the claims	1 hr	Shlok
3	Create a UI for modifying current claims with new information.	3 hr	Shlok
4	Process the data through the backend and update it in the database	2 hr	Shlok
5	Send an updated notification to the staff location	1 hr	Shlok
6	Implement tests to make sure data gets modified properly	2 hr	Shlok

Acceptance Criteria:

- Given the student can see their past claim requests, they should have an option to view and modify their requests.
- Given the student fills in the details, they should be validated to check if they pass basic criteria and see that there are some differences in the request.
- Given that the student updates their claim, a new notification should be sent out to the concerned users.

User Story #8

As a user, I want to see the top items being reported as lost (based on trends).

#	Description	Estimated Time	Owner
1	Add an Analytics page for students	2 hr	Daivik
2	Page should list the most frequent item categories lost	2 hr	Daivik
3	Implement backend for updating counter dynamically every time a new item from that category is added	2 hr	Daivik
4	Add a field in the items database for category	1 hr	Daivik

Acceptance Criteria:

- There should be an Analytics page for Students.
- This page should display all the most frequent categories of items being lost.
- The database should store the items based on category
- The database should be updated every time a new item for that category is uploaded
- The display counter should be updated every time a new item for that category is added.

User Story 9

As a user, I want to see a list of all previously claimed items.

#	Description	Estimated Time	Owner
1	Add a toggle button to the filter pane	1 hr	Daivik
2	Display the items that were claimed when the toggle button is activated	3 hr	Daivik
3	Add an attribute in the items database that checks whether the item was claimed or not.	1 hr	Daivik
4	Add unit tests for the new attribute in the database.	1 hr	Daivik

Acceptance Criteria:

- There should be a toggle button on the feed.
- The toggle button should be placed in the left filter pane.
- By toggling the button, the feed will only show the items that have been claimed in the past.
- By not toggling the button, the feed will not show the items that were already claimed.
- The database should store past claimed items and mark them as claimed

User Story #10

As a user, I want to have my recently claimed items to be automatically registered in my account.

#	Description	Estimated Time	Owner
1	Add ‘my registered items’ button to user profile	30 mins	Shreya
2	Create UI for student user to see their pre-registered items	2 hr	Shreya
3	When an item status is changed to ‘claimed’ for a particular user, change the ‘is_pre_registered’ column for that particular item to ‘1’	1 hr	Shreya
4	Create backend API endpoints to return user’s pre-registered item details from the database	3 hr	Shreya
5	Create unit tests to test correct functionality of user viewing pre-registered items	2 hr	Shreya

Acceptance Criteria:

- The ‘my registered items’ button should be visible and accessible on the user’s profile page.
- When a user clicks the ‘my registered items’ button, they should be redirected to a page displaying a list of their pre-registered items.
- The UI for viewing pre-registered items should clearly display each pre-registered item belonging to the logged-in student user, including relevant details such as item name, description, and status (e.g: claimed, in dispute, etc).
- The backend should correctly retrieve and return the pre-registered item details from the database

User Story #11

As a user, I want to start a claim dispute form on wrongly claimed items.

#	Description	Estimated Time	Owner
1	Add ‘Dispute Claim’ button to the ‘item status’ page	30 mins	Shreya
2	Create UI for the claim dispute form	3 hr	Shreya
3	Implement backend logic and API endpoints to receive dispute form data from the user	4 hr	Shreya
4	Validate form data types	2 hr	Shreya
5	Store this dispute information in the DB	2.5 hr	Shreya

Acceptance Criteria:

- The ”Dispute claim” button should be visible and accessible on the individual item’s status view page for each item.
- When the “Dispute claim” button is clicked by the user, they should be redirected to the ‘dispute claim’ form.
- The student user should be able to fill out necessary fields in the dispute form such as reason for dispute, additional comments, etc.
- In the ‘dispute claim’ form, if any of the fields are left blank by the student user, an error message should be displayed prompting them to fill out all fields.
- The student user should be able to submit a ‘dispute claim’ form only after filling out all of the fields in the form.

Staff-side:

User Story #12

As a staff member, I want to view a portal of all pending claim requests.

#	Description	Estimated Time	Owner
1	Create UI to view all pending claim requests	3 hr	Shlok
2	Implement backend to create a listing of all pending claims	2 hr	Shlok
3	Create UI to view individual claim requests for further info	3 hr	Shlok
4	Implement backend to pull individual claims and display them in the individual view.	2 hr	Shlok
5	Provide option buttons in the individual view	1 hr	Shlok

Acceptance Criteria:

- As a staff user, I want to be able to view all pending claim requests from the students
- Given the staff selects a particular claim to review, an individual claim view should open up with all the details of the claim from the student.
- Given the claim request makes sense / does not make sense, there should be an option to accept it / deny it.

User Story #13

As a staff member, I want to see a list of all unclaimed items that have been in the system for more than 1 week.

#	Description	Estimated Time	Owner
1	Add button to staff item listing page	1 hr	Daniel
2	Trim results to only items older than one week	1 hr	Daniel
3	Sort remaining items by age	1 hr	Daniel
4	Add button for user to reverse sorting	1 hr	Daniel
5	Item sorting should respect pins	2 hr	Daniel

Acceptance Criteria:

- Given there is a page for staff to view items, there should be a button to filter out items younger than a week old
- Given the button filters out older items, there should be a toggle to allow the filter to be reversed and show only items younger than a week old
- Given the filters work, the items should be sorted in ascending age
- Given the items are sorted by ascending age, there should be a button to allow the sorting to be reversed and display items in descending age
- Given the sorting works, the page should respect the order of pinned items there may be

User Story #14

As a staff user, I want to log in with a secure staff-only authentication

#	Description	Estimated Time	Owner
1	Add a staff sign up form	1 hr	Daivik
2	Add an approval system for staff accounts	2 hr	Daivik
3	Add a staff login form	1 hr	Daivik
4	Add a secure staff only database	2 hr	Daivik
5	Add tests for the staff only database	1hr	Daivik
6	Implement the backend for the entire staff authentication process	4 hr	Daivik

Acceptance Criteria:

- There should be an exclusive staff sign up only form.
- Given that there is an exclusive staff sign up only page, there should be a way for the accounts administrator to approve staff sign ups.
- There should be a staff login only page
- There should be a secure staff exclusive database to store the staff login credentials.
- Given that the staff member has logged in, it should redirect to the staff's item feed.

User Story #15

As a staff member, I want to print a receipt for students when they claim a lost item.

#	Description	Estimated Time	Owner
1	Add link/button to approved listings that navigates a UI that has basic student and item information	1 hr	Anitej
2	Create the UI for the basic student and item information	3 hr	Anitej
3	Have the UI display the basic student and item information	4 hr	Anitej
4	Allow for staff user to export to pdf for printing	3 hr	Anitej

Acceptance Criteria:

- Given staff has navigated to a listing that has been approved for student claim, there should be a link/button to print the listing with student information
- The link/button should navigate to a UI that shows what information will be exported to pdf to be printed
- The UI should show basic student and item information
- The staff user should be able to export this UI to pdf for printing

User Story #16

As a staff member, I want to see reports on lost and found activity (number of items, most common locations).

#	Description	Estimated Time	Owner
1	Add an Analytics page	2 hr	Daivik
2	Implement a counter for number of items that have been claimed	1 hr	Daivik
3	Implement a counter for number of items that have been unclaimed	1 hr	Daivik
4	List the locations where items most frequently went missing	1 hr	Daivik
5	List the most common category of items that were lost	1 hr	Daivik

Acceptance Criteria:

- There should be a separate staff Analytics page that displays all of these statistics.
- Given that there is an Analytics page, it should display the number of items that have been claimed.
- Given that there is an Analytics page, it should display the number of items that have been unclaimed.
- Given that there is an Analytics page, it should display the locations where items were frequently lost
- Given that there is an Analytics page, it should display the most common category of items that were lost/found.

User Story #17

Staff User: As a staff member, I want to be able to accept, deny or request further information for claims and claim requests.

#	Description	Estimated Time	Owner
1	Create form to “deny” claim/ claim request	4 hr	Anitej
2	Create button to “accept” claim (items not flagged high-value)	2 hr	Anitej
3	Create button to “review” claim request, presenting the student-user’s claim request to staff	2 hr	Anitej
4	Create button to request more info/schedule meeting with student that submitted a claim request	4 hr	Anitej

Acceptance Criteria:

- Given that the staff-user can view all submitted claims and claim requests, they should be able to “deny” a claim/claim request and provide rationale.
- Given that the staff-user can view all submitted claims and claim requests, they should be able to “accept” a claim (not-flagged)
- Given that the staff-user can view all submitted claims and claim requests, they should be able to “accept” a claim request(flagged as high value)
- Given that the staff-user can view all submitted claims and claim requests, they should be able to “review” a claim request
- Given that the staff-user can view all submitted claims and claim requests, they should be able to “request more info” or schedule a meeting with a student that submitted a claim request
- All outcomes should notify student

User Story #18

As a staff member, I want to be able to “release” an item after filling out the required release form

#	Description	Estimated Time	Owner
1	Create button on each claim/claim request that links to a release form	0.5 hr	Mihika
2	Create UI for release form containing Student name, mobile, image of ID	2 hr	Mihika
3	Create option to edit release form	1 hr	Mihika
4	Update database containing info about claimed items, student id	3 hr	Mihika
5	Remove claim request from pending requests listing	2 hr	Mihika

Acceptance Criteria:

- When a student comes to pick up their item, staff-user should be able to navigate to their claim/claim request and pull up a release form.
- Given that the release form loads, staff-user should be able to enter student name, mobile number, and submit an image of student id
- Given that the staff-user submits the release form, the database should be updated with item meta data and student id image.
- Given that the release form submission was successful, the claim/claim request should be removed form the listing of claim/claim requests pending pick up.

User Story #19

As a staff member, I want to categorise found items in bulk (for batch uploads).

#	Description	Estimated Time	Owner
1	Create new staff input form for generalised low-value items	3 hr	Anitej
2	Create generalized categories for low-value items	3 hr	Anitej
3	Allow items to be inputted in bulk	4 hr	Anitej

Acceptance Criteria:

- When inputting items, a staff member should be able to redirect to a form to input a large number of low-value items
- The staff member should be able to pick from categories for low-value items
- The staff member should be able to create listings in bulk with no description

User Story #20

As a staff member, I want to use filters to quickly find items that have been reported as lost by a specific student.

#	Description	Estimated Time	Owner
1	Page to look at lost item reports	3 hr	Daniel
2	Table on page that contains date, name or reporter, description, and location	3 hr	Daniel
3	Buttons to filter and sort	1 hr	Daniel
4	Button on staff page to redirect to page	1 hr	Daniel
5	Button to return to homepage	1 hr	Daniel

Acceptance Criteria:

- Given the staff have a homepage, there is a button to take the user to the page
- Given the button on the staff page exists, there is a page to go to
- Given the page exists, the staff is presented with a table that contains all lost item reports with relevant information like the date, location lost, and name of reporter
- Given the table is there, there are buttons to sort by a particular column
- Given the table is there, there is a search bar to find items reported as lost by a searched name
- Given the page exists, there is a button to return to the home page

Total Hours Per Person for Sprint 2:

#	**Name**	**Total Time**
1	Anitej	33 hr
2	Daivik	30 hr
3	Daniel	36 hr
4	Mihika	32 hr
5	Shlok	30 hr
6	Shreya	30 hr

Gantt Chart for Sprint 2:

Included at the end.

Remaining Backlog:

Functional Requirements:

Student Side -

1. ~~As a user, I want to have the option to create an account with my email.~~
2. ~~As a user, I want my details (preferred name and pronouns) to be assigned to my account.~~
3. ~~As a user, I want to have a reset password option.~~
4. ~~As a user, I want to be able to deactivate my account.~~
5. ~~As a user, I want to view a list of all lost items on the platform.~~
6. ~~As a user, I want to be able to search listings with keywords.~~
7. ~~As a user, I want to filter lost items by category.~~
8. ~~As a user, I want to see detailed information about a lost item.~~
9. ~~As a user, I want to see images of the found items uploaded by desk staff.~~
10. ~~As a user, I want to be able to filter items by location (where they were lost or found).~~
11. ~~As a user, I want to be able to filter by the date the item was lost.~~
12. ~~As a user, I want to flag an item that I believe to be mine for further verification.~~
13. ~~As a user, I want to view the current status of my flagged items (claimed, in dispute, etc.).~~
14. ~~As a user, I want to pre-input details like description, date, and location about my lost item to help match found items.~~
15. ~~As a user, I want to receive a notification when an item matching my description of a lost item is found.~~
16. As a user, I want to be able to post a template of my lost item request on social media (if time allows).
17. ~~As a user, I want to claim my item through a verification process (e.g., providing additional details or proof).~~
18. As a user, I want to report an item I have found.
19. As a user, I want to see a map of the nearby help desks to drop off the items I found.
20. As a user, I want to view the items I have reported as found.
21. ~~As a user, I want to modify my claim on a lost item in case of error.~~
22. As a user, I want to get a map view of the locations where lost items were reported.
23. ~~As a user, I want to see the top items being reported as lost (based on trends).~~
24. As a user, I want to be able to “register” a personal item: generate a unique identifier/QR code for a personal item that maps directly to my user account and item details.
25. As a user, I want to be able to save the QR as a printable sticker and be provided steps on printing at a designated location (WALC).
26. As a user, I want to be able to view all, add and delete my registered items.
27. ~~As a user, I want to see a list of all previously claimed items.~~
28. ~~As a user, I want to have my recently claimed items to be automatically registered in my account.~~

29. ~~As a user, I want to start a claim dispute form on wrongly claimed items.~~
30. As a user, I want to be able to communicate with the help desk in case of disputes.
31. As a user, I want to leave a review or feedback on the lost and found service.
32. As a user, I want to be able to view a listing of items moved to the surplus store for sale (if time allows).
33. As a user, I want to be able to buy items online from the surplus store (if time allows).

Front Desk/Staff Side -

1. ~~As a staff member, I want to log in with a secure staff-only authentication.~~
2. ~~As a staff member, I want to enter a new found item into the system with details (location, description).~~
3. ~~As a staff member, I want to upload an image of a found item to the system.~~
4. ~~As a staff member, I want to tag the item with specific characteristics (e.g., size, brand, color).~~
5. As a staff member, I would like to generate searchable keyword descriptions for listings using visual search AI.
6. ~~As a staff member, I want to transfer an item to the central lost and found facility in the system.~~
7. ~~As a staff member, I want to view a portal of all pending claim requests.~~
8. ~~As a staff user, I want to be able to modify pre-existing items in the system.~~
9. ~~As a staff member, I want to flag items for special attention (e.g., high-value items).~~
10. As a staff member, I want to log disputes between students over claimed items.
11. ~~As a staff member, I want to see a list of all unclaimed items that have been in the system for more than 1 week.~~
12. As a staff member, I want to track the history of the movement of an item across help desks.
13. ~~As a staff member, I want to archive items that have been claimed or moved to the surplus store for sale.~~
14. As a staff member, I want to manage listings on the online surplus store marketplace (if time allows).
15. ~~As a staff member, I want to print a receipt for students when they claim a lost item.~~
16. ~~As a staff member, I want to see reports on lost and found activity (number of items, most common locations).~~
17. As a staff member, I want to be able to scan pre registered items for easy verification of ownership.
18. ~~As a staff member, I want to send alerts to students who have claimed items for pickup.~~
19. ~~As a staff member, I want to collect students' identification before releasing an item.~~
20. ~~As a staff member, I want to delete or edit incorrect entries in the system.~~
21. As a staff member, I want to categorize found items in bulk (for batch uploads).
22. ~~As a staff member, I want to use filters to quickly find items that have been reported as lost by a specific student.~~
23. As a staff member, I want to see the real-time status of all items at my location.

24. As a staff member, I want to see which items have been flagged for review by students.
25. As a staff member, I want to receive automated reminders to transfer items to the central lost and found.
26. As a staff member, I want to view activity logs for each item to ensure accountability.
27. As a staff member, I want to post weekly unclaimed item listings to social media (if time allows).

Non-functional Requirements:

- **Architecture and Performance**

  Database: sqlite3
  Frontend: React
  Backend: Flask

  BoilerTrack will use a modern web architecture with a React frontend and a Flask backend. We will use SQLite for data storage. The frontend built with React will provide a user-friendly interface for the reporting and claiming of lost items. The Flask backend, powered by Python, will handle the core application logic, manage data processing, API requests, and integrations with the database. The lightweight SQLite database will store all item-related data and user-account related data. RESTful APIs will ensure smooth communication between the front and back ends, handling tasks like user authentication and claim management. This architecture will provide an efficient and scalable solution to centralize lost and found operations across Purdue University.

- **Security**

  For BoilerTrack, we need to securely store user accounts. We plan on using Supabase for security of user account information. We will use OAuth with authentik. University Staff will have special permissions to add/update/delete items that regular users will not have.

- **Usability**

  The UI should be minimalist, showing only essential information like search fields and item lists. We want the main page to load in 500 ms. Navigation must be easy, with users able to browse, search, and report items within a few clicks. Consistent design across all pages is key. Feedback should include clear success and error messages. Search should allow filtering by date, location, and item type, with fast, relevant results. A “Help” or FAQ section will address common questions. The app must stay under 1 GB by limiting image sizes and unnecessary code, ensuring no lag with efficient API calls and database management. We want to load 100 concurrent instances.

- **Hosting/Deployment**

  We will be hosting in containers for easy reproducibility using docker-compose on a free Oracle vps. This will allow us to bring up all the services (like the container for authentik, and its supporting entourage) for the application in one command.

STORY NUMBER	TASK DESCRIPTION	OWNER	START DATE	DUE DATE	WEEK 4				WEEK 5					WEEK 6				
					T	W	R	F	M	T	W	R	F	M	T	W	R	F
1	**Student User: As a student user, I want to have a reset password option.**																	
	Add verification for if account exists on the backend and send confirmation or error message	Daniel	10/15	10/21		■												
	Add password reset token generation and store it in the database	Daniel	10/15	10/21		■	■											
	Create page to input new password	Daniel	10/15	10/21		■												
	Create unit test to check correct functionality of password reset by checking database for relevant updates	Daniel	10/15	10/21				■										
	Add verification of token and update password in the database	Daniel	10/15	10/21				■										
2	**Student User: As a student user, I want to be able to deactivate my account.**																	
	Create a page that gives the option to close an account	Daniel	10/14	10/21	■													
	Add confirmation prompts in the form of “Please type <email/username> to confirm you want to delete your account”	Daniel	10/15	10/21		■												
	Add final password verification if a user wants to delete their account	Daniel	10/15	10/21		■												
	Server should verify that the logged in user is the account that will be removed	Daniel	10/15	10/21		■												
	Job should be run to remove accounts marked as deleted in bulk during low traffic hours	Daniel	10/15	10/21			■											
	Server should remove the account from the database by marking it as deleted	Daniel	10/15	10/21			■											
3	**Student User: As a user, I want to view the current status of my flagged items (claimed, in dispute, etc.)**																	
	Add a “view claim requests” button to the student user’s profile page	Shreya	10/15	10/15	■													
	Create UI for student user to see items for which they have submitted a claim request	Shreya	10/15	10/16	■	■												
	Fetch user’s claim requests details from the database	Shreya	10/17	10/17			■											
	Create backend API endpoints to return details about user’s claim requests from the database	Shreya	10/17	10/17			■											
	Create unit tests to test correct functionality of user viewing claim requests	Shreya	10/18	10/18				■										
4	**Student User: As a user, I want to pre-input details like description, date, and location about my lost item to help match found items.**																	
	Create UI form for users to be able to enter item details.	Mihika	10/15	10/16	■	■												
	Connect/send item data to be stored in database via backend routing	Mihika	10/16	10/17		■	■											
	Create edit,delete option to modify form details	Mihika	10/17	10/17			■											
	Create ui to view all lost item requests	Mihika	10/18	10/18				■										
	Add lost item request to other requests associated to user in db	Mihika	10/18	10/18				■										
5	**Student User: As a user, I want to receive a notification when an item matching my description of a lost item is found.**																	
	Create UI for notification tab	Mihika	10/21	10/21					■									
	Create button for user to go to item view of item that matches description	Mihika	10/21	10/22					■	■								
	Update status of request in lost-item-requests listing (pending, match-found, claimed, etc)	Mihika	10/23	10/23							■							
	Update lost-item-request status in database (pending, match-found, claimed, etc)	Mihika	10/24	10/24								■						
	Create backend functionality to check for “item matches” between items in the database and lost-item-requests logged b	Mihika	10/24	10/25								■	■					
6	**Student User: As a user, I want to claim my item through a verification process (e.g., providing additional details or proof).**																	
	Add “Submit Claim Request” button to the item detail page and a “Submit Proof” UI form	Shlok	10/15	10/15	■													
	Accept all relevant and valid data from the user	Shlok	10/16	10/16		■												
	Submit the data to the backend to store for review	Shlok	10/16	10/17		■	■											
	Send a notification to the staff location where the item was found about the claim request	Shlok	10/16	10/18		■	■	■										
7	**Student User:As a user, I want to modify my claim on a lost item in case of error.**																	
	The student should be able to view all their past claim requests	Shlok	10/28	10/28										■				
	Add a modify request button to the claims	Shlok	10/28	10/28										■				
	Create a UI for modifying current claims with new information.	Shlok	10/29	10/30											■	■		
	Process the data through the backend and update it in the database	Shlok	10/30	10/31												■	■	
	Send an updated notification to the staff location	Shlok	10/31	10/31													■	
	Implement tests to make sure data gets modified properly	Shlok	11/1	11/1														■
8	**Student User: As a user, I want to see the top items being reported as lost (based on trends).**																	
	Add an Analytics page for students	Daivik	10/15	10/21	■													
	Page should list the most frequent item categories lost	Daivik	10/15	10/21	■	■												
	Implement backend for updating conuter dynamically every time a new item from that category is added	Daivik	10/15	10/21		■	■											
	Add a field in the items database for category	Daivik	10/15	10/21			■	■										
9	**Student User: As a user, I want to see a list of all previously claimed items.**																	
	Add a toggle button to the filter pane	Daivik	10/21	10/25					■									
	Display the items that were claimed when the toggle button is activated	Daivik	10/21	10/25					■	■								
	Add an attribute in the items database that checks whether the item was claimed or not.	Daivik	10/21	10/25						■	■							
	Add unit tests for the new attribute in the database.	Daivik	10/21	10/25							■	■	■					
10	**Student User: As a user, I want to have my recently claimed items to be automatically registered in my account..**																	
	Add ‘my registered items’ button to user profile	Shreya	10/21	10/21					■									
	Create UI for student user to see their pre-registered items	Shreya	10/21	10/22					■	■								
	When an item status is changed to ‘claimed’ for a particular user, change the ‘is_pre_registered’ column for that particula	Shreya	10/22	10/23						■	■							
	Create backend API endpoints to return user’s pre-registered item details from the database	Shreya	10/23	10/24							■	■						
	Create unit tests to test correct functionality of user viewing pre-registered items	Shreya	10/25	10/25									■					
11	**Student User: As a user, I want to start a claim dispute form on wrongly claimed items.**																	
	Add ‘Dispute Claim’ button to the ‘item status’ page	Shreya	10/28	10/28										■				
	Create UI for the claim dispute form	Shreya	10/28	10/29										■	■			
	Implement backend logic and API endpoints to receive dispute form data from the user	Shreya	10/29	10/30											■	■		
	Validate form data types	Shreya	10/31	11/1													■	
	Store this dispute information in the DB	Shreya	10/31	11/1													■	■
12	**Staff User: As a staff member, I want to view a portal of all pending claim requests.**																	
	Create UI to view all pending claim requests	Shlok	10/21	10/22					■	■								
	Implement backend to create a listing of all pending claims	Shlok	10/22	10/22						■								
	Create UI to view individual claim requests for further info	Shlok	10/23	10/24							■	■						
	Implement backend to pull individual claims and display them in the individual view.	Shlok	10/24	10/24								■						
	Provide option buttons in the individual view	Shlok	10/25	10/25									■					
13	**Staff User: As a staff member, I want to see a list of all unclaimed items that have been in the system for more than 1 week.**																	
	Add button to staff item listing page	Daniel	10/21	10/23					■									
	Trim results to only items older than one week	Daniel	10/21	10/23					■	■								
	Sort remaining items by age	Daniel	10/21	10/23					■	■	■							
	Add button for user to reverse sorting	Daniel	10/21	10/23						■	■							
	Item sorting should respect pins	Daniel	10/21	10/23							■							
14	**Staff User: As a staff user, I want to log in with a secure staff-only authentication**																	
	Add a staff sign up form	Daivik	10/15	10/31					■					■				
	Add an approval system for staff accounts	Daivik	10/15	10/31					■	■				■	■			
	Add a staff login form	Daivik	10/15	10/31						■	■				■	■		
	Add a secure staff only database	Daivik	10/15	10/31							■	■				■	■	
	Implement the backend for the entire staff authentication process	Daivik	10/15	10/31								■	■				■	■
	Add tests for the staff only database	Daivik	10/15	10/31									■					■
15	**Staff User: As a staff member, I want to print a receipt for students when they claim a lost item.**																	
	Add link/button to approved listings that navigates a UI that has basic student and item information	Anitej	10/15	10/21	■													
	Create the UI for the basic student and item information	Anitej	10/15	10/21	■	■												
	Have the UI display the basic student and item information	Anitej	10/15	10/21		■	■											
	Allow for staff user to export to pdf for printing	Anitej	10/15	10/21			■	■										
16	**Staff User: As a staff member, I want to see reports on lost and found activity (number of items, most common locations).**																	
	Add an Analytics page	Daivik	10/28	11/1										■				
	Implement a counter for number of items that have been claimed	Daivik	10/28	11/1										■	■			
	Implement a counter for number of items that have been unclaimed	Daivik	10/28	11/1											■	■		
	List the locations where items most frequently went missing	Daivik	10/28	11/1												■	■	
	List the most common category of items that were lost	Daivik	10/28	11/1													■	■
17	**Staff User: As a staff member, I want to be able to accept, deny or request further information for claims and claim requests.**																	
		Anitej	10/21	10/22					■	■								
	Create button to “accept” claim (items not flagged high-value)	Anitej	10/23	10/24							■	■						
	Create button to “review” claim request, presenting the student-user’s claim request to staff	Anitej	10/24	10/24								■						
	Create button to request more info/schedule meeting with student that submitted a claim request	Anitej	10/25	10/25									■					
18	**As a staff member, I want to be able to “release” an item after filling out the required release form**																	

	Create button on each claim/claim request that links to a release form	Mihika	10/28	10/28										🟪				
	Create UI for release form containing Student name, mobile, image of ID	Mihika	10/28	10/28										🟪				
	Create option to edit release form	Mihika	10/29	10/29											🟪			
	Update database containing info about claimed items, student id	Mihika	10/30	10/31												🟪	🟪	
	Remove claim request from pending requests listing	Mihika	10/31	11/1													🟪	🟪
19	**As a staff member, I want to categorise found items in bulk (for batch uploads).**																	
	Create new staff input form for generalised low-value items	Anitej	10/28	10/29										🟦	🟦			
	Create generalized categories for low-value items	Anitej	10/29	10/30											🟦	🟦	🟦	
	Allow items to be inputted in bulk	Anitej	10/30	11/1													🟦	🟦
20	**Staff User: As a staff member, I want to use filters to quickly find items that have been reported as lost by a specific student.**																	
	Page to look at lost item reports	Daniel	10/24	10/31								🟥	🟥	🟥				
	Table on page that contains date, name or reporter, description, and location	Daniel	10/24	10/31										🟥	🟥	🟥	🟥	
	Buttons to filter and sort	Daniel	10/24	10/31										🟥	🟥	🟥	🟥	
	Button on staff page to redirect to page	Daniel	10/24	10/31								🟥	🟥					
	Button to return to homepage	Daniel	10/24	10/31								🟥	🟥					