Sprint 3 Review

Date: 04/10/2020

Sprint: 3

Scrum Master: Vincent Villaflores (s3728807)

Development Team: Meng Kheang Leng (s3704080), Chhayhy Kourn (s3699618), Hue Phuong Le (s3687477), William Bossen (s3658961)

Sprint Goals

The goal of the third sprint is to begin work on three features including login/logout for three different user roles, allowing customers to register an account and displaying all bookings for admin users, giving them the ability to manage their bookings (confirm or cancel the pending bookings). Moreover, we will continue work on the deployment automatically using Docker in AWS. As mentioned in the last sprint, we will also ensure the requirements of the Definition of Done (DoD) have been met. The product backlog will be updated accordingly.

Status Overview

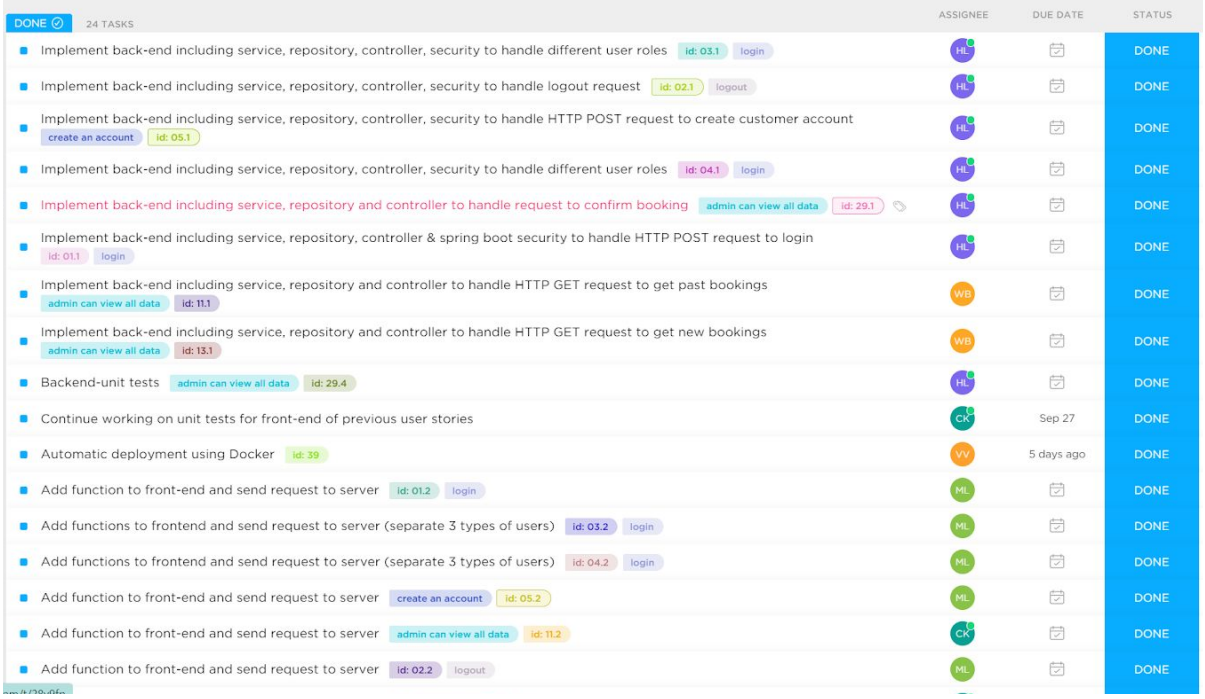
A table below shows the information of product backlog items will be discussed.

Task ID	Task name	User story	Status	Demo
1.1	Implement back-end including service, repository, controller & spring boot security to handle HTTP POST request to login	As an admin, I want to login so that I can manage all company account information and ensure all details are correct	Done	Yes
1.2	Add function to front-end and send request to server		Done	Yes
1.3	Front-end unit tests		Done	Yes
1.4	Back-end unit tests		Done	Yes
2.1	Implement back-end including service, repository, controller, security to handle logout request	As a logged in user, I want to be able to logout of my own account, so that I can prevent unauthorised people from accessing and/or modifying data with my account	Done	Yes
2.2	Add function to front-end and send request to server		Done	Yes
2.3	Front-end unit tests		Done	Yes

2.4	Back-end unit tests		Done	Yes
3.1	Implement back-end including service, repository, controller, security to handle different user roles	As a worker, I want to login, so that I can see my dashboard and keep track of all my work	Done	Yes
3.2	Add functions to frontend and send request to server (separate 3 types of users)		Done	Yes
3.3	Add front-end unit tests to separate 3 types of users		Done	Yes
3.4	Add back-end unit tests to separate 3 types of users		Done	Yes
4.1	Implement back-end including service, repository, controller, security to handle different user roles	As a customer, I want to login so that i can book an appointment with my details	Done	Yes
4.2	Add functions to frontend and send request to server (separate 3 types of users)		Done	Yes
4.3	Add front-end unit tests to separate 3 types of users		Done	Yes
4.4	Add back-end unit tests to separate 3 types of users		Done	Yes
5.1	Implement back-end including service, repository, controller, security to handle HTTP POST request to create customer account	As a customer, I want to register an account with all relevant information, so that my data will be saved to the database and every time I book an appointment, I do not need to re-enter them again	Done	Yes
5.2	Add function to front-end and send request to server		Done	Yes
5.3	Front-end unit tests		Done	Yes
5.4	Back-end unit tests		Done	Yes
11.1	Implement back-end including service, repository and controller to handle GET request to get past bookings	As an admin, I want be able to see the past bookings so that I can see which of my employees have been working the hardest	Done	Yes
11.2	Add function to front-end and send request to server		Done	Yes
11.3	Front-end unit tests		Done	Yes
11.4	Back-end unit tests		Done	Yes

13.1	Implement back-end including service, repository and controller to handle GET request to get new bookings	As an admin, I want to see all new bookings, so that I can stay up-to-date and manage my business efficiently	Done	Yes
13.2	Add function to front-end and send request to server		Done	Yes
13.3	Front-end unit tests		Done	Yes
13.4	Back-end unit tests		Done	Yes
29.1	Implement back-end including service, repository and controller to handle request to confirm booking	As an admin, I want to be able to confirm or cancel the new booking, so that I can know the customer made an appropriate booking or not	Done	Yes
29.2	Add function to front-end and send request to server		Done	Yes
29.3	Front-end unit tests		Done	Yes
29.4	Back-end unit tests		Done	Yes
39	Automatic deployment using Docker	N/A	Done	Yes

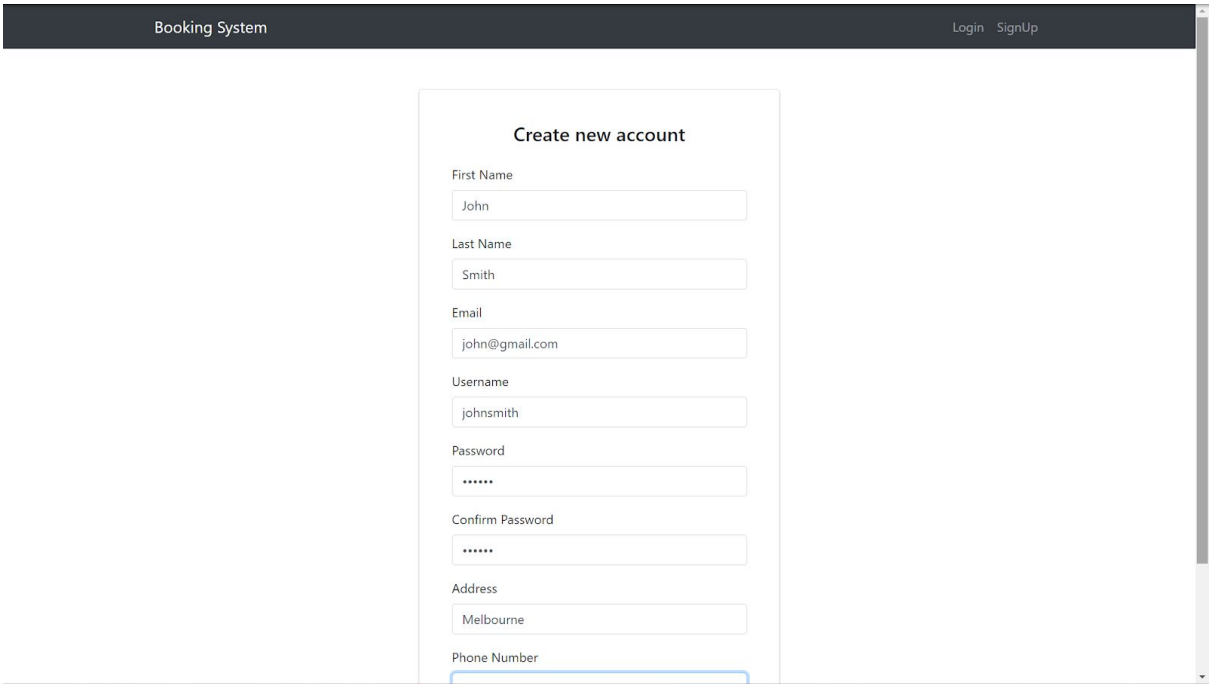
A screenshot of Clickup:



Screenshots

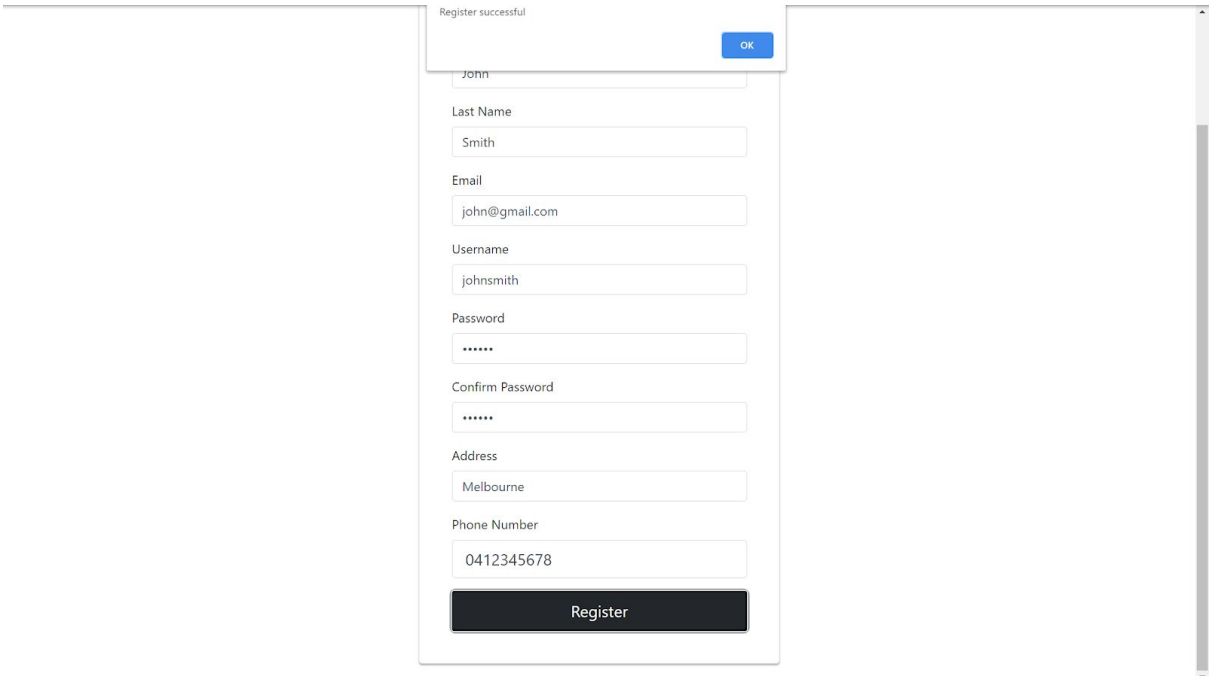
- **Register an account**

By entering the link “localhost:3000/register” or select the “SignUp” button in the navigation bar, the user will be redirected to the page displaying a “Create new account” form for them to fill in their details and click the “Register” button.

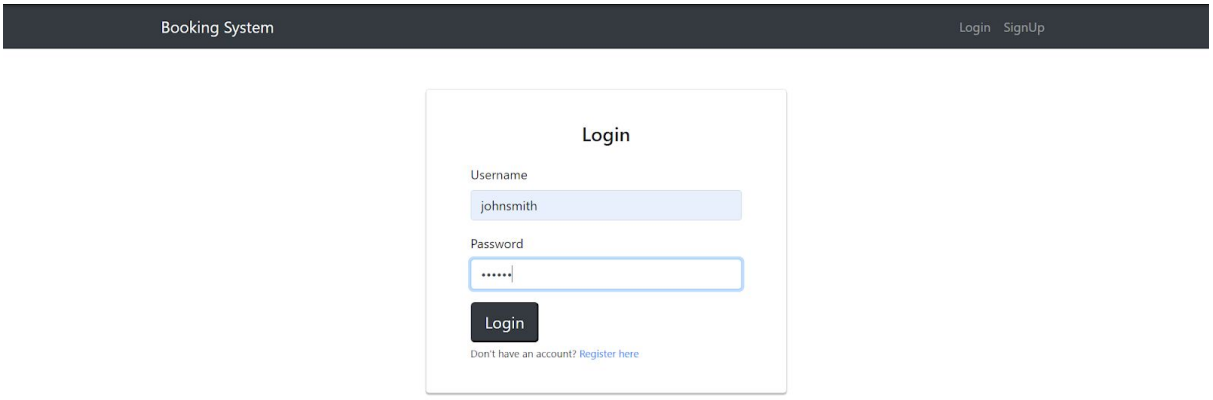


If all the information is valid, satisfy all the constraints including:
- Valid email
- Password has at least 6 character & confirm password has to match password
- Phone number is 10 digit
- Username length is between 3 and 21

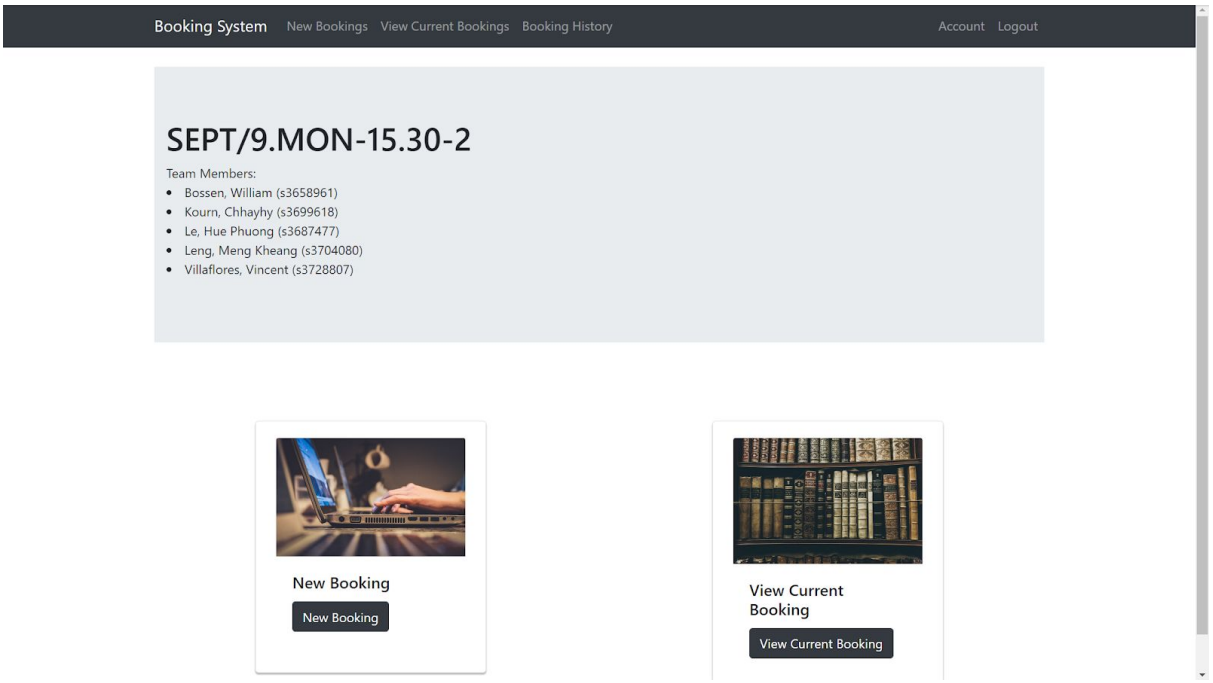
The message “Register successful” will appear to notify the user and by clicking on the “OK” button, they are directed to the login page.



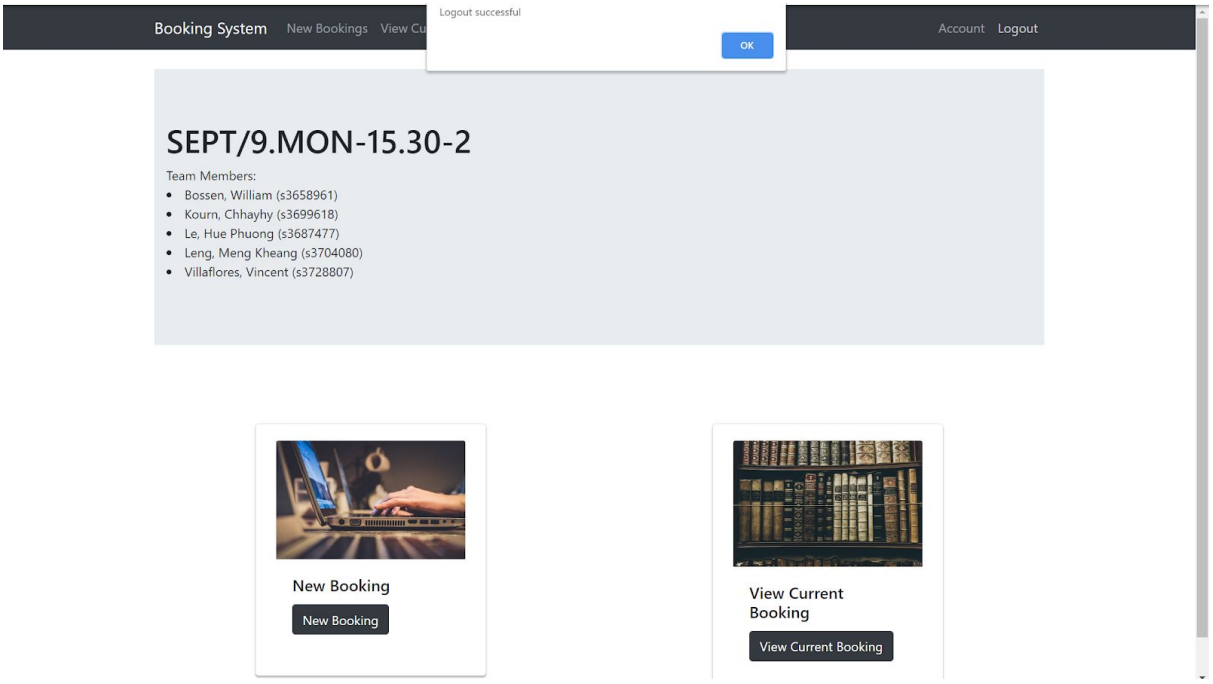
- **Login/Logout**

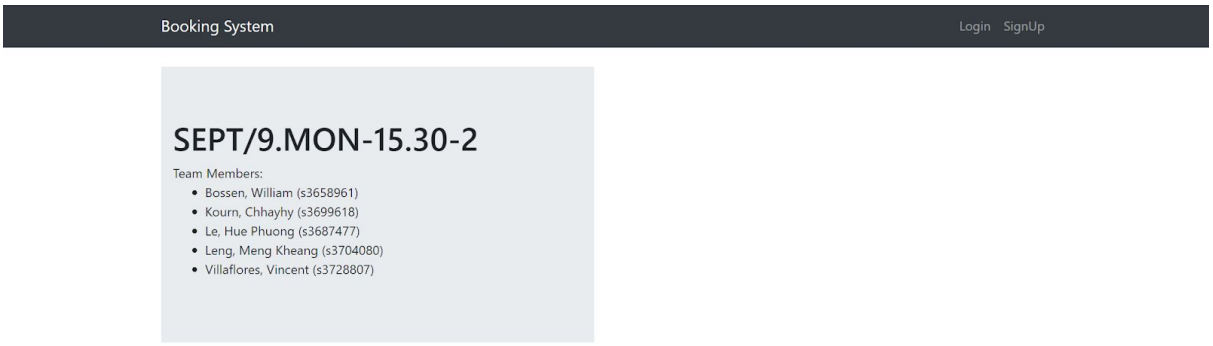


After input all correct username, password and click on “Login” button, user is redirected to their appropriate homepage. The below is the homepage for customers. As we have 3 different user roles, there will be four homepage, including the homepage for guest users.

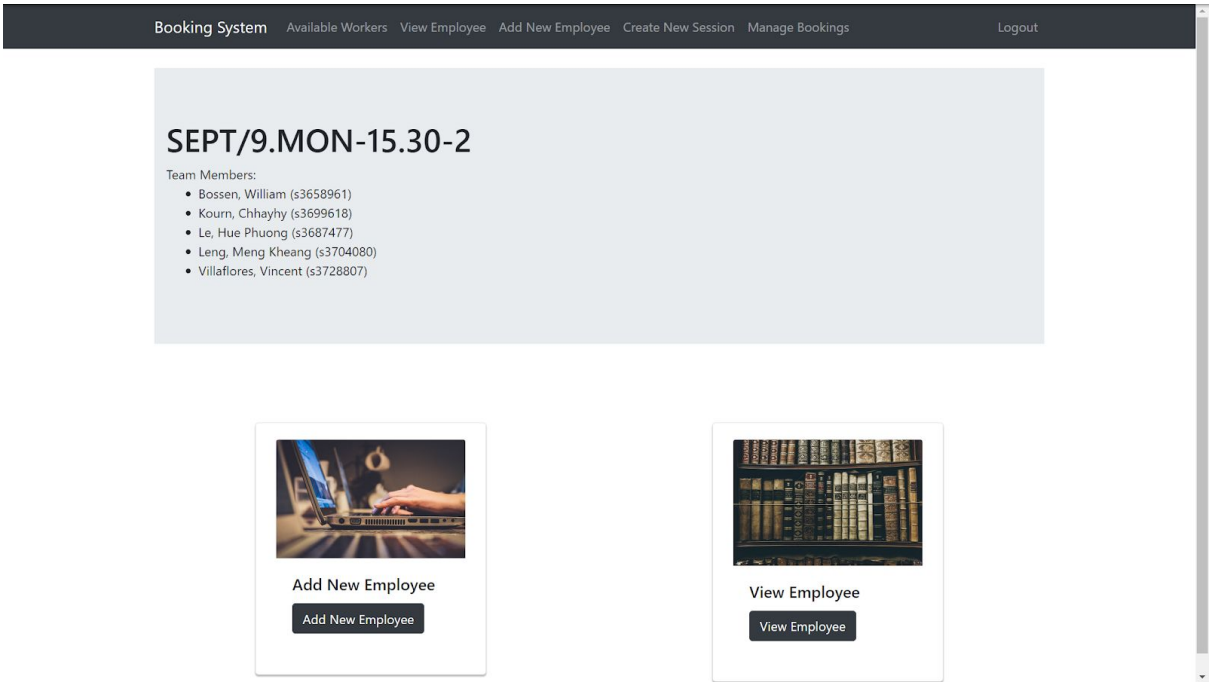


Selecting the “Logout” button will display the message “Logout successfully” and the customer is redirected to the guest homepage.



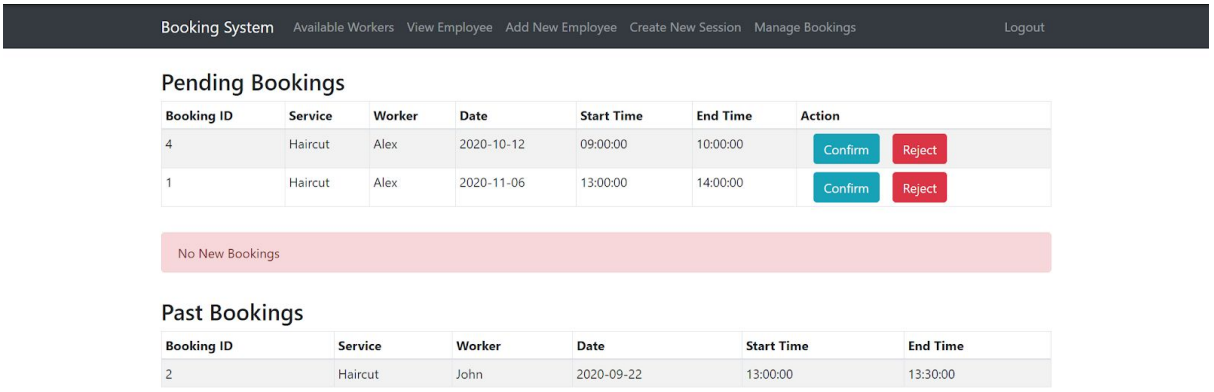


This is the homepage for admin accounts.

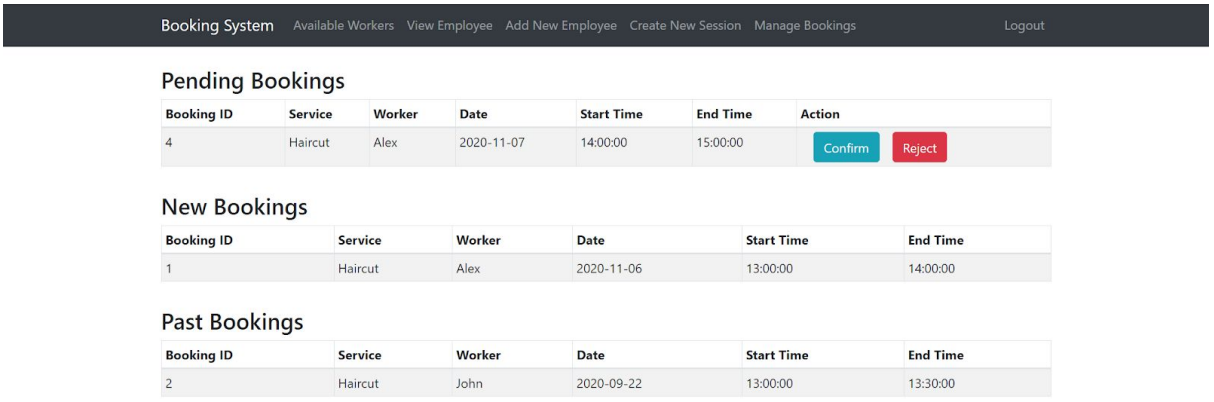


- **Admin viewing and managing bookings**

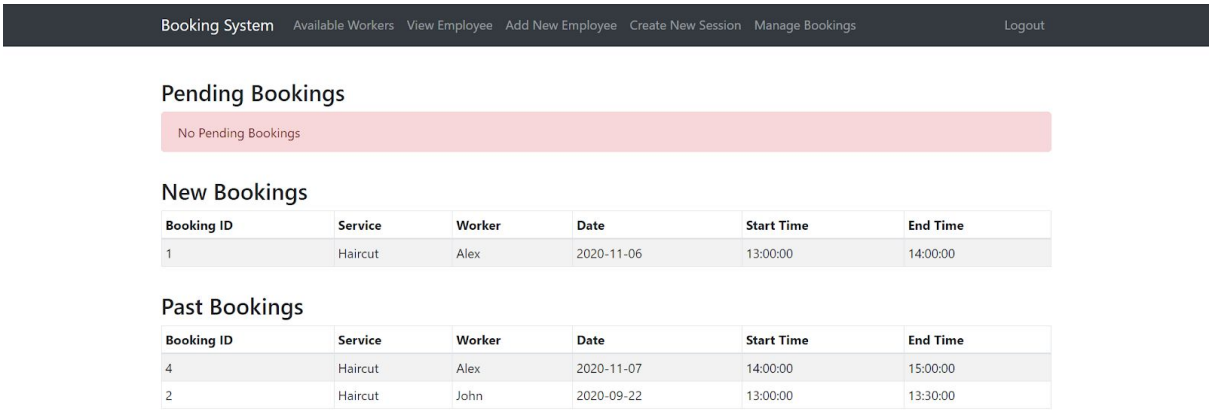
Selecting the “Manage Bookings” button in the navigation bar, the admin will be redirected to the page displaying all their bookings including new bookings, past bookings and pending bookings.



If admin “Confirm” the booking and click on the “Confirm” button, the booking status will be changed to “CONFIRM” and it will be displayed in “New Bookings”.

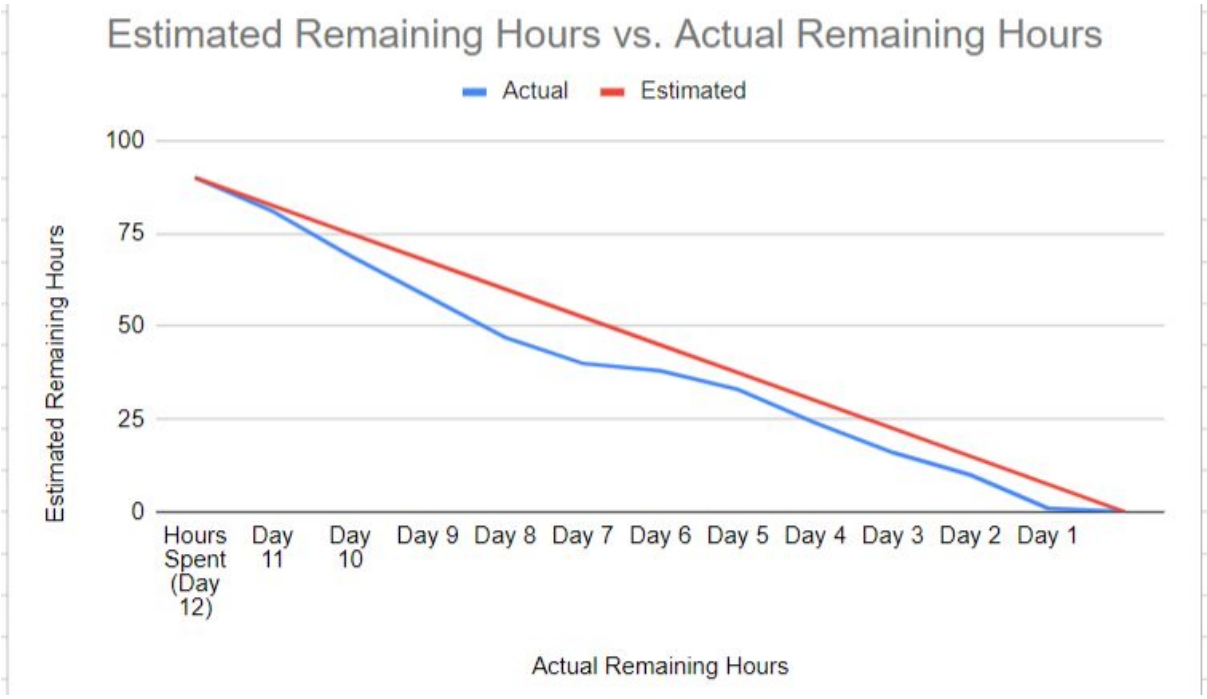


Otherwise, it admin “Reject” a booking, the booking will be cancelled and displayed in “Past Bookings”



Sprint Statistics

Task	Start Hours	Hours Spent (Day 12)	Day 11	Day 10	Day 9	Day 8	Day 7	Day 6	Day 5	Day 4	Day 3	Day 2	Day 1	Total Hours
1.1	3	2	1	0	0	0	0	0	0	0	0	0	0	3
1.2	3	0	0	2	1	0	0	0	0	0	0	0	0	3
1.3	2	0	0	0	0	2	0	0	0	0	0	0	0	2
1.4	2	0	0	0	0	2	0	0	0	0	0	0	0	2
2.1	2	0	0	2	0	0	0	0	0	0	0	0	0	2
2.2	2	0	0	0	0	1	1	0	0	0	0	0	0	2
2.3	1	0	0	0	0	0	0	1	0	0	0	0	0	1
2.4	1	0	0	0	0	0	0	1	0	0	0	0	0	1
3.1	3	3	0	0	0	0	0	0	0	0	0	0	0	3
3.2	3	0	3	0	0	0	0	0	0	0	0	0	0	3
3.3	2	0	0	1	1	0	0	0	0	0	0	0	0	2
3.4	2	0	0	0	1	1	0	0	0	0	0	0	0	2
4.1	3	1	2	0	0	0	0	0	0	0	0	0	0	3
4.2	3	0	2	1	0	0	0	0	0	0	0	0	0	3
4.3	2	0	0	0	2	0	0	0	0	0	0	0	0	2
4.4	2	0	0	0	2	0	0	0	0	0	0	0	0	2
5.1	4	0	0	0	0	0	0	3	1	0	0	0	0	4
5.2	7	0	0	0	0	0	0	0	3	2	2	0	0	7
5.3	3	0	0	0	0	0	0	0	0	0	0	3	0	3
5.4	3	0	0	0	0	0	0	0	0	0	0	2	1	3
11.1	2	1	1	0	0	0	0	0	0	0	0	0	0	2
11.2	2	0	0	2	0	0	0	0	0	0	0	0	0	2
11.3	1	0	0	0	1	0	0	0	0	0	0	0	0	1
11.4	1	0	0	0	1	0	0	0	0	0	0	0	0	1
13.1	2	0	0	2	0	0	0	0	0	0	0	0	0	2
13.2	2	0	0	0	2	0	0	0	0	0	0	0	0	2
13.3	1	0	0	0	0	1	0	0	0	0	0	0	0	1
13.4	1	0	0	0	0	0	1	0	0	0	0	0	0	1
29.1	4	0	0	0	0	0	0	0	2	2	0	0	0	4
29.2	7	0	0	0	0	0	0	0	3	2	2	0	0	7
29.3	3	0	0	0	0	0	0	0	0	0	0	3	0	3
29.4	3	0	0	0	0	0	0	0	0	0	2	1	0	3
39	8	2	3	1	0	0	0	0	0	2	0	0	0	8
Actual Remaining Hours	90	81	69	58	47	40	38	33	24	16	10	1	0	90
Estimated Remaining Hours	90	82.5	75	67.5	60	52.5	45	37.5	30	22.5	15	7.5	0	



The third sprint had a total of 33 tasks, completed over 12 days. We spent a total of 90 hours to achieve this sprint. The red line depicts the ideal situation, with the estimated hours distributed to show a steady steep on the graph. In comparison to this, we spent our hours well, as displayed by the blue line.

It is clearly shown that during this sprint, the estimated hours were always higher than anticipated, with the difference between the estimated remaining hours and the actual remaining hours, being approximately 5 hours each day. As a result, the performance of all members in this sprint was extremely good and we completed the sprint even before the deadline.