Group Project Log

Group Name	Team 19

Group Members	Vivek Patel, Rishi Patel, Meet Patel, Rutvik Patel, Tirth Patel, Paras Patel

Deliverable	**TakeOnRent** project proposal and front-end code (via **GitLab**)

Group Member Name	Work Done (%)
Vivek Patel	16.6 %
Rushi Patel	16.6 %
Meet Patel	16.6 %
Rutvik Patel	16.6 %
Tirth Patel	16.6 %
Paras Patel	16.6 %
Total	**100%**

Project Proposal

Project Group 6

TakeOnRent: The application TakeOnRent would allow the users to conveniently navigate and find the product that would be required for a specific interval of time without buying the product.

Members and Contributors

Vivekkumar Patel
B00874162
vv662564@dal.ca

Rushi Patel
B00886157
rs397441@dal.ca

Meet Patel
B00899516
mt631537@dal.ca

Paras Patel
B00890745
pr777796@dal.c

Tirth Patel
B00890745
tr326604@dal.ca

Rutvik Patel
B00897762
rt994397@dal.ca

Faculty of Computer Science

Dalhousie University

15 June, 2022

Content

1. Project Background

Our group have discussed and finalized the project – TakeOnRent. On this platform, any user can egister and rent a product. The purpose of this application is for the users to conveniently navigate and find the product that would be required for a specific interval of time without buying the product. TakeOnRent also gives an option to take insurance on the products while renting for better security and to manage risk.

The application has several features including user management, preference management (keeping the user's preference in mind based on the interested categories), community forum (any user of the application can view the posts related to the products based on the category with latest updates), messaging service (user can message owner regarding the product to rent), reviews (Owner's review and the product's review), analytics (statistics of the user's behavior to rent), billing and payment, insurance (option to opt for insurance for the selected product), product catalogue, cart management, product setup, and coupon management (option to add coupons for discounts to the rental products from the owner).

1.1 Project Objectives

It is quite evident that most people would like to have a flexible and cost-effective method to obtain a product when they need it and return the product when the product is no longer required. The application - TakeOnrent is a rental application where we aim to achieve a similar objective.

The main objective of the application is to provide users with an easy-to-use rental service. Users would like to rent based on multiple factors. Renting allows you to avoid unnecessary investment, maintenance, and other expenses [1]. Renting allows you to get the equipment you need when you need it, which is more efficient than buying and seeing your assets decay in value while they aren't in use [1]. TakeOnRent offers the right product for the job, resulting in increased efficiency [1]. Product is maintained regularly, and rental assures that the product is of excellent quality and reliability while on-site [1]. Renting gives the user flexibility by allowing them to rent for a short, medium, or long period [1].

1.2 Live Project URL

Our application can be accessed at: https://takeonrent.netlify.app/

Our GitLab repository can be found at: https://git.cs.dal.ca/rspatel/csci_5709_project_group_19.git

2. Application Details

2.1 Target User Insight

This website targets four main groups of individuals: Businessmen, Sales Associates, International students and insurance agents. The businessmen mostly are the business owners who own multiple equipment and products and want to put that equipment on rent so that others can utilize those types of equipment without upfront investment. Hence those business owners can take advantage of the “TakeOnRent” platform to generate extra income by putting all their equipment and products on rent. The sales associates are those individuals who sell equipment. A sales associate can utilize the “TakeOnRent” website by providing equipment on rent along with the selling of new equipment to earn additional money. International students are the other target users who utilize “TakeOnRent” for renting some appliances rather than buying new ones.

The user interface of TakeOnRent has been designed to be as user-friendly as possible so that it won’t require several training sessions for target users. Here we are assuming that the target users have basic computer skills. In addition, they are also familiar with the general structure of online shopping websites, so there will be no need for any additional training.

2.2 Brand Attributes

TakeonRent is a Web Application whose primary focus is Home Owners, Small Businesses, and Building design and construction workers (Mechanical, Plumbing, and Electrical). This application aims to provide the essential equipment needed to repair or maintain the construction, landscaping, or cleaning requirements of buildings, businesses, houses, commercials, etc. Now, the base requirement for keeping clients' trust to use this application is that they get all the detailed information related to their rentals, including what they bought and the user guide for that equipment (Videos of how to use, manuals). The most crucial part is the payment processing, where people put their credit card or other financial institution's details. So we have designed this seamless web app to navigate and protect with an Authentication gateway for payment processing (we can take help from existing secured gateways). Apart from that, we have feedback from the client about their services and continuous improvement in the service, which will make them feel like we care for them.

2.3 Competitive Landscape

Take on Rent is a website that helps people to Find Household utility products to take on rent at an affordable price and their convenience. Our website will help customers to find household products such as washing machines, fridges, gardening machines, tools, accessories, and gaming products such as play stations and x-box.

Take on Rent satisfies the need of people who wants products for temporary use at a very reasonable price and on-demand. The platform is also useful for the people who have extra products and want to rent them to earn some side income. The platform has an insurance tool, which allows users to get insurance on a product before taking it for rent.

United rental is our existing competitor.

The details that distinguish us are listed below:

Product insurance:

Users will be able to get insurance on the product before taking on rent and claim the insurance in case of damage.

2.4 Project Scope

2.4.1 Project Goals:

1. Provide an opportunity to all users to rent equipment which they are owning and are not in active use. Instead of keeping equipment idle make some money from renting equipment.
2. Help users to connect with such users who own equipment and want to give them for rent.
3. Encourage users to give equipment they own on the tent.
4. Enable users to continue with the work which is pending due to unavailability of the types of equipment.
5. Provide a medium for users to make connections with like-minded users.
6. The platform enables the user to be the owner and user of the product at the same time.
7. Allow users to share feedback on the products and owners.

2.4.2 Existing Features:

Profile Management: The web application has profile management options using that both owner and consumer can log in to the portal and browse through various products. This functionality allows only authorized users to change their password and profile details.

Equipment Management: Users can add products which they want to put on rent. Moreover, users can view all the products which users have created.

Search Products: Users can search different products by name of the product, category, and price of the product.

Billing and Invoice: Once all the products are added to the cart, the user can proceed with the payment process. Users can have various user details necessary for the payment.

2.4.3 New features:

Preference Management: In the existing platform there is no feature by which users can have only those products in which they are interested to have on rent. By using this feature users can provide the various categories of items in which they are interested. Next onwards when the user login into the account, the user will get items according to this preference.

Notification Management: Using this feature can send messages to the owner of the equipment. The recipient user will get an email notification containing message details. This will initiate communication between two parties interested in providing products on rent and taking products on rent.

Product Verification: Once the product is made in the system product will be shown to users in pending status. The user will verify all the details and will publish the product to put it on rent.

Privacy and User management: Using this feature authorize users can change their account details, preferences for the different product category and change the password.

2.5 Information Architecture

2.5.1 Proposed Sitemap

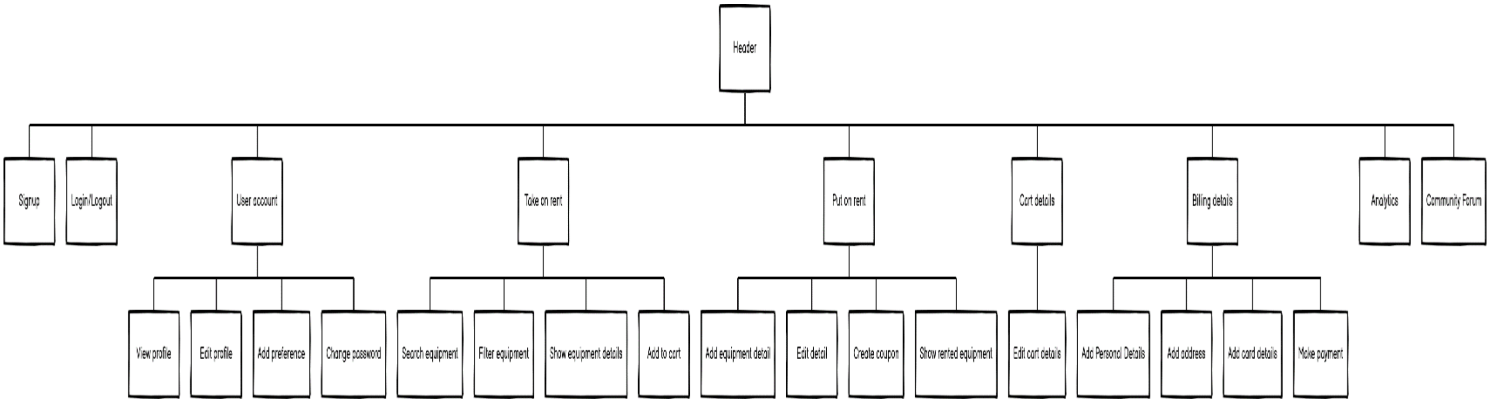


Figure 1: Header Sitemap

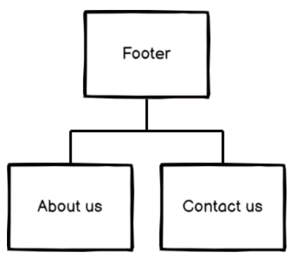


Figure 2: Footer Sitemap

2.5.2 Proposed Wireframes

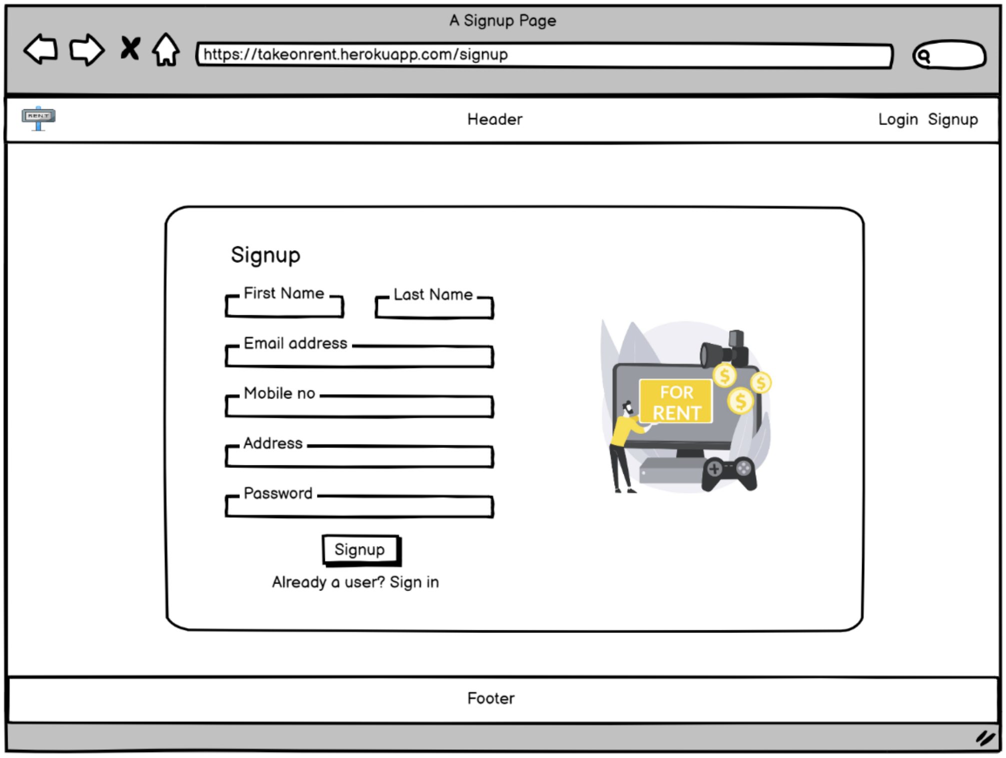


Figure 3: User signup Low Fidelity Template

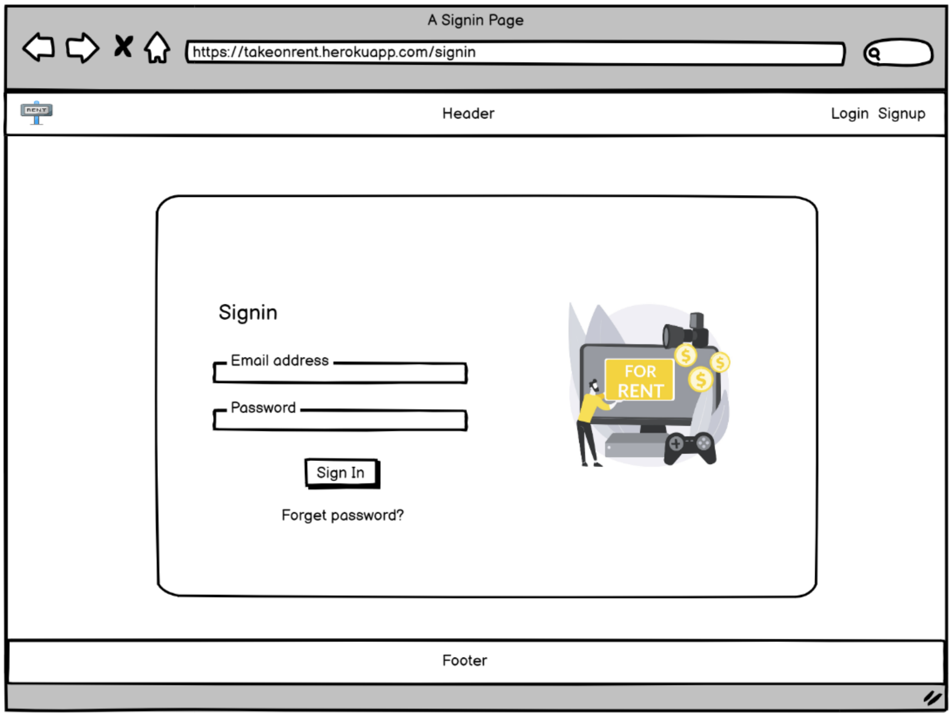


Figure 4: User sign in Low Fidelity Template

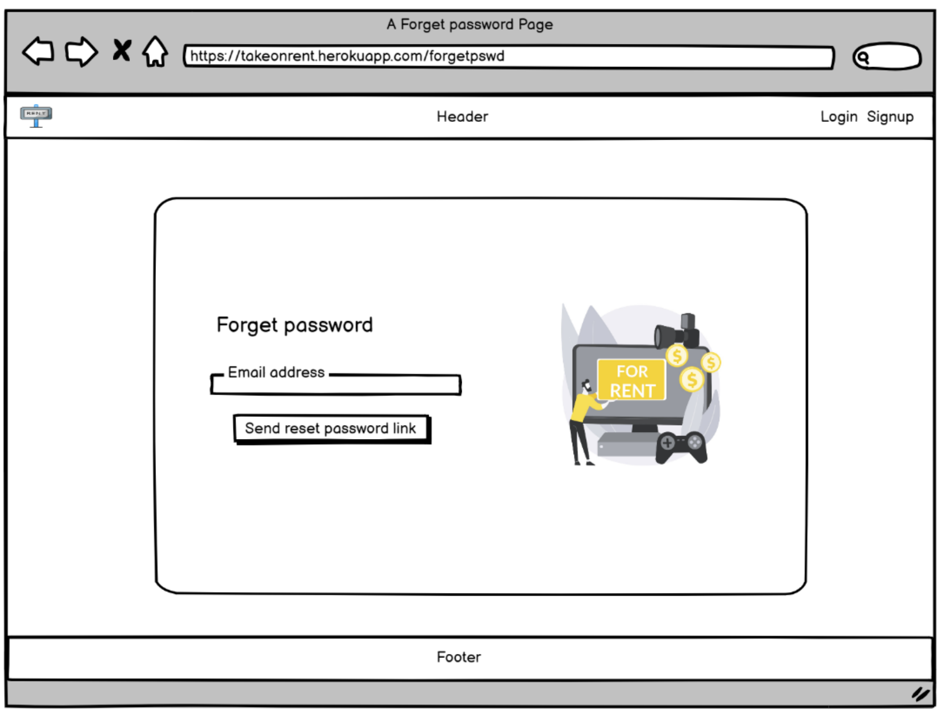


Figure 5: Forget Password Low Fidelity Template

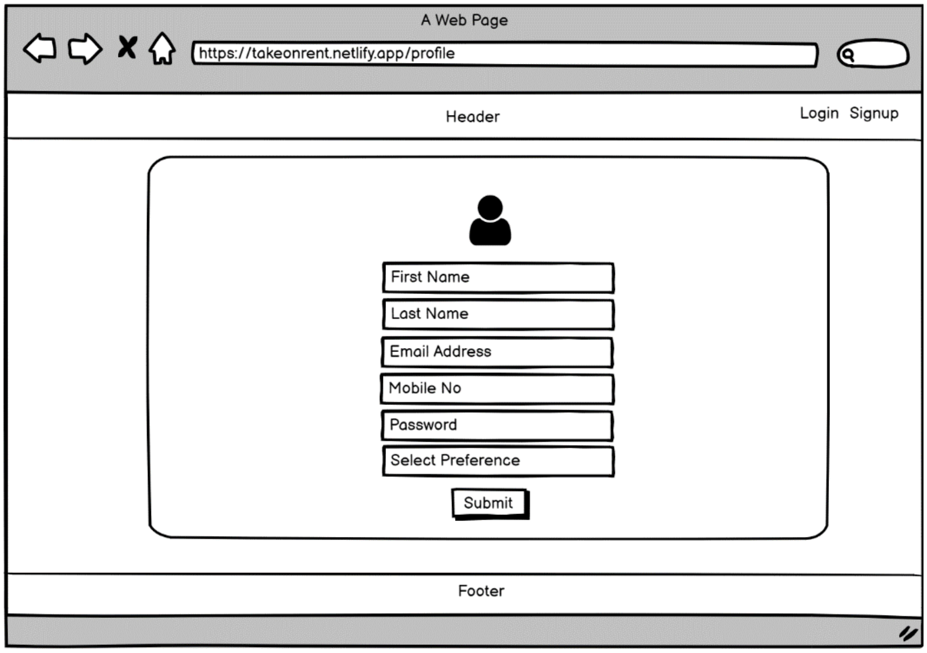


Figure 6: User Profile Low Fidelity Template

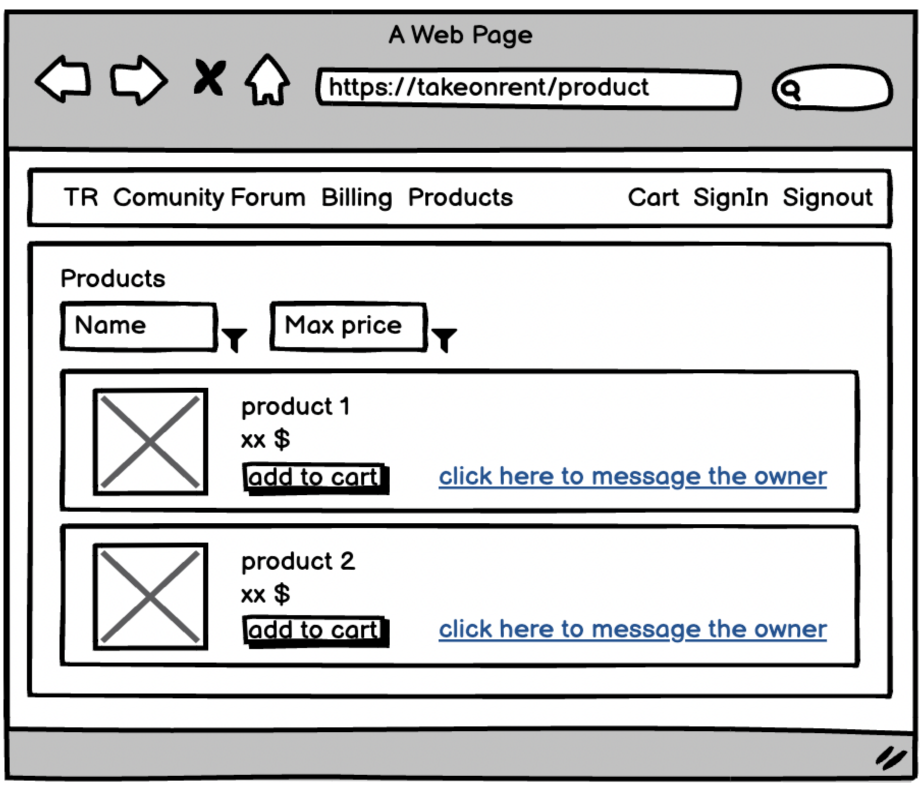


Figure 7: Cart Management Low Fidelity Template

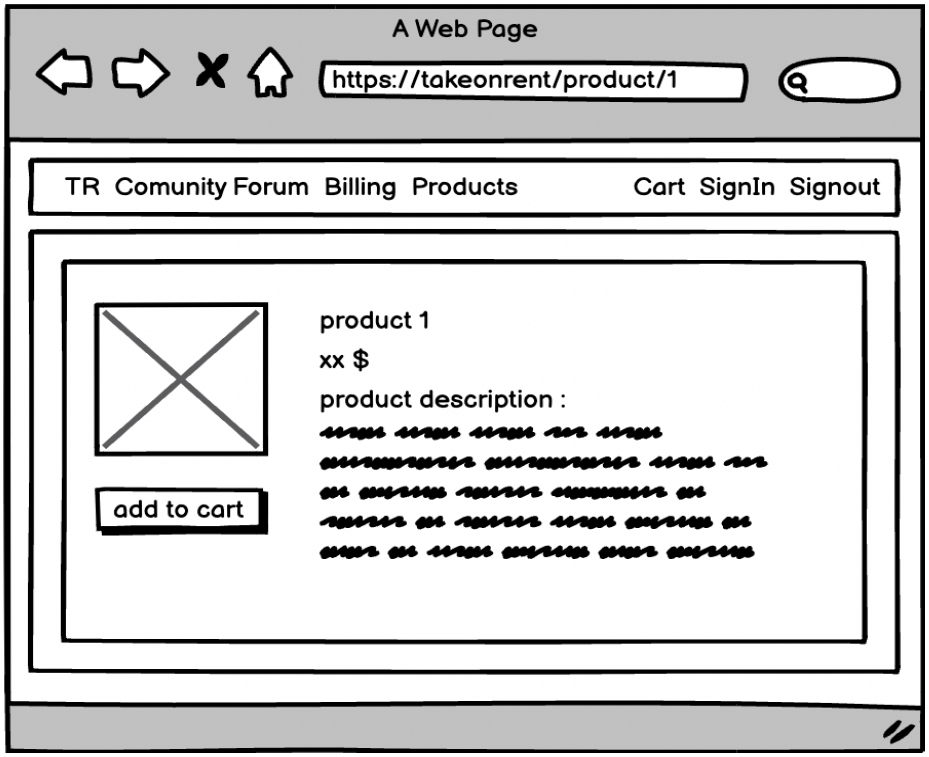


Figure 8: Single Product Low Fidelity Template

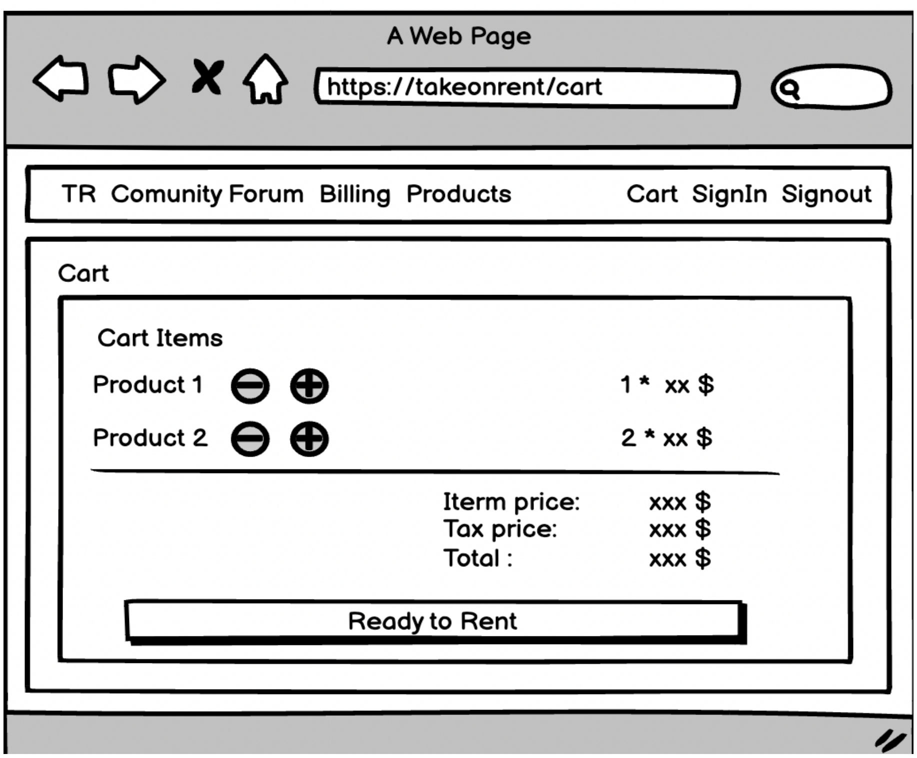


Figure 9: Cart Low Fidelity Template

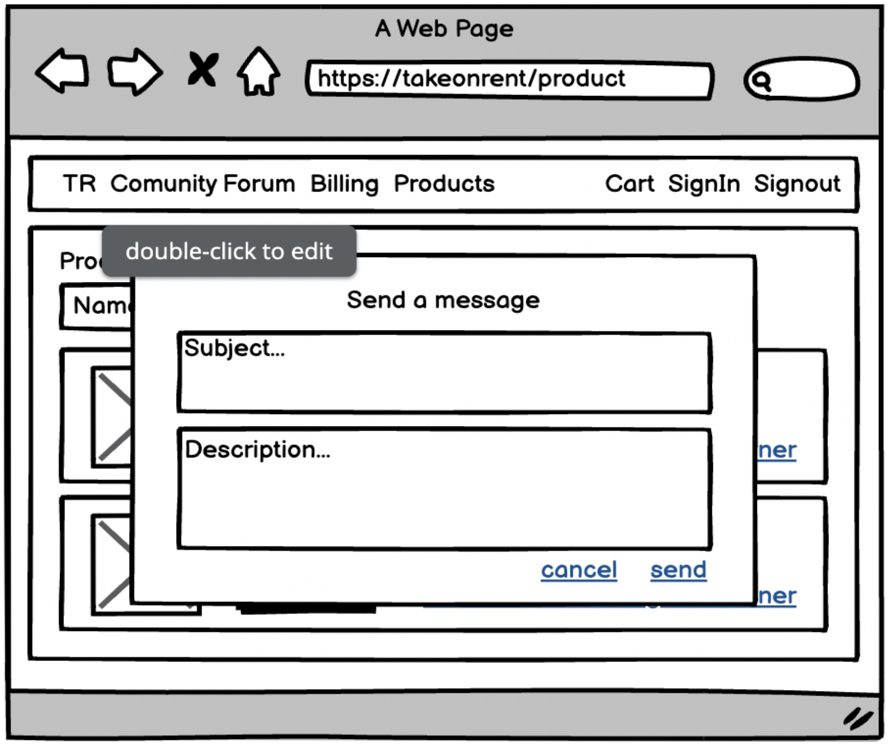


Figure 10: Send Message Low Fidelity Template

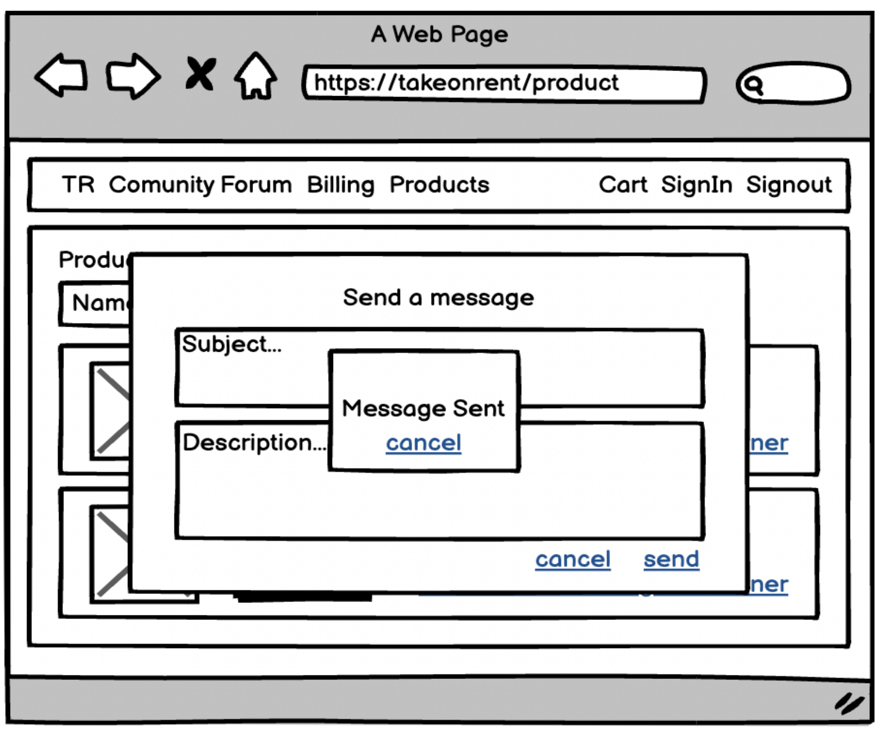


Figure 11: Message sent successfully Low Fidelity Template

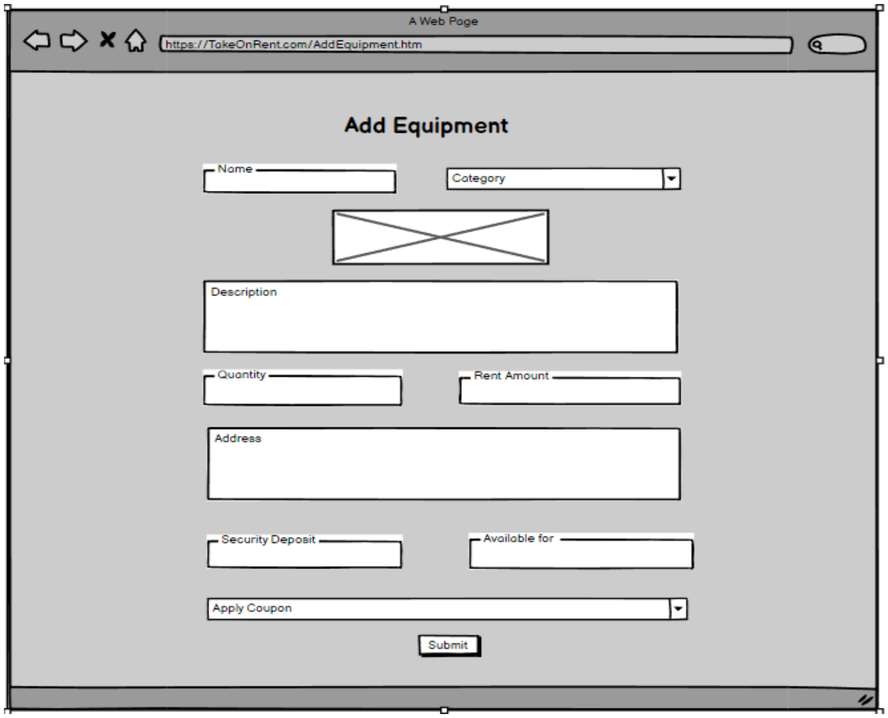


Figure 12: Add Equipment Low Fidelity Template

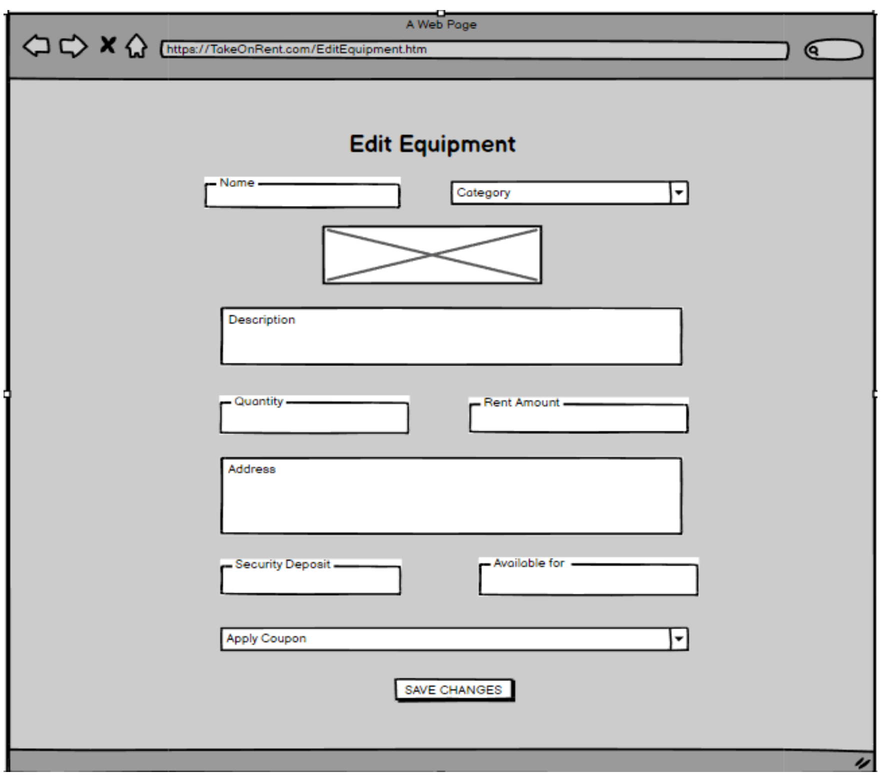


Figure 13: Edit Equipment Low Fidelity Template

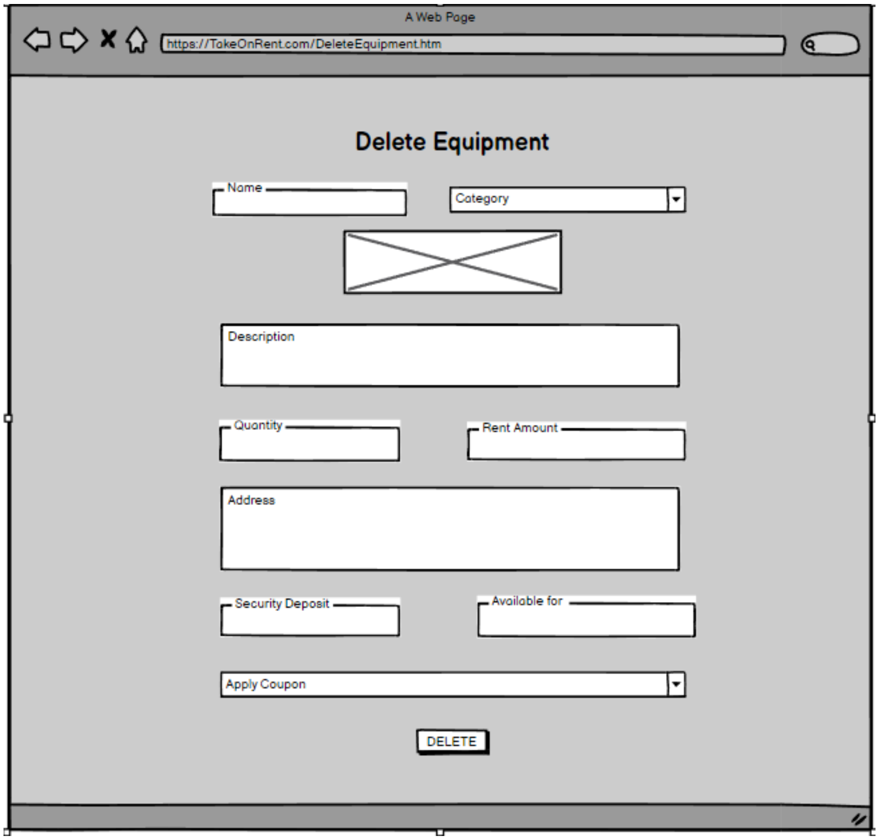


Figure 14: Delete Equipment Low Fidelity Template

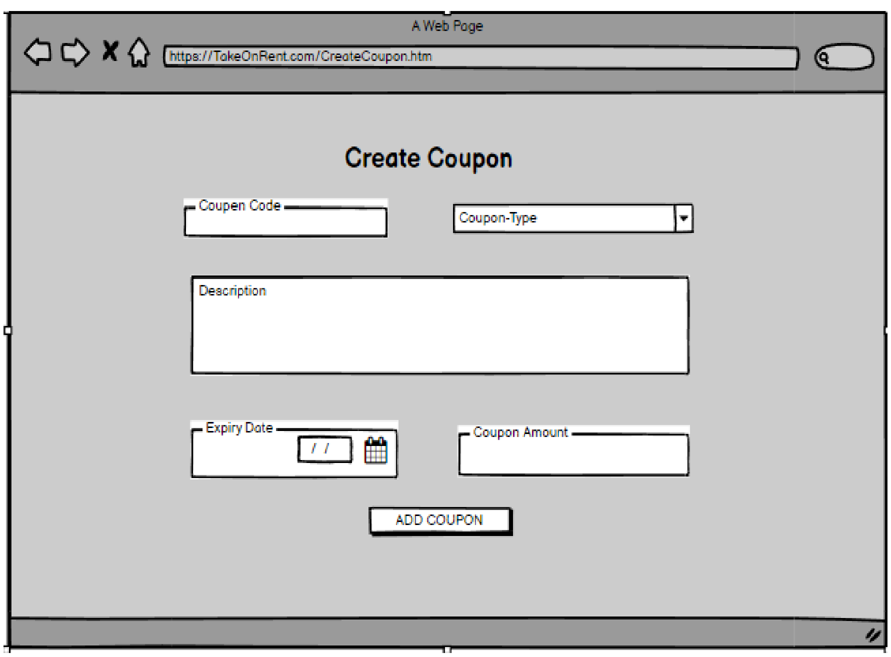


Figure 15: Create Coupon Low Fidelity Template

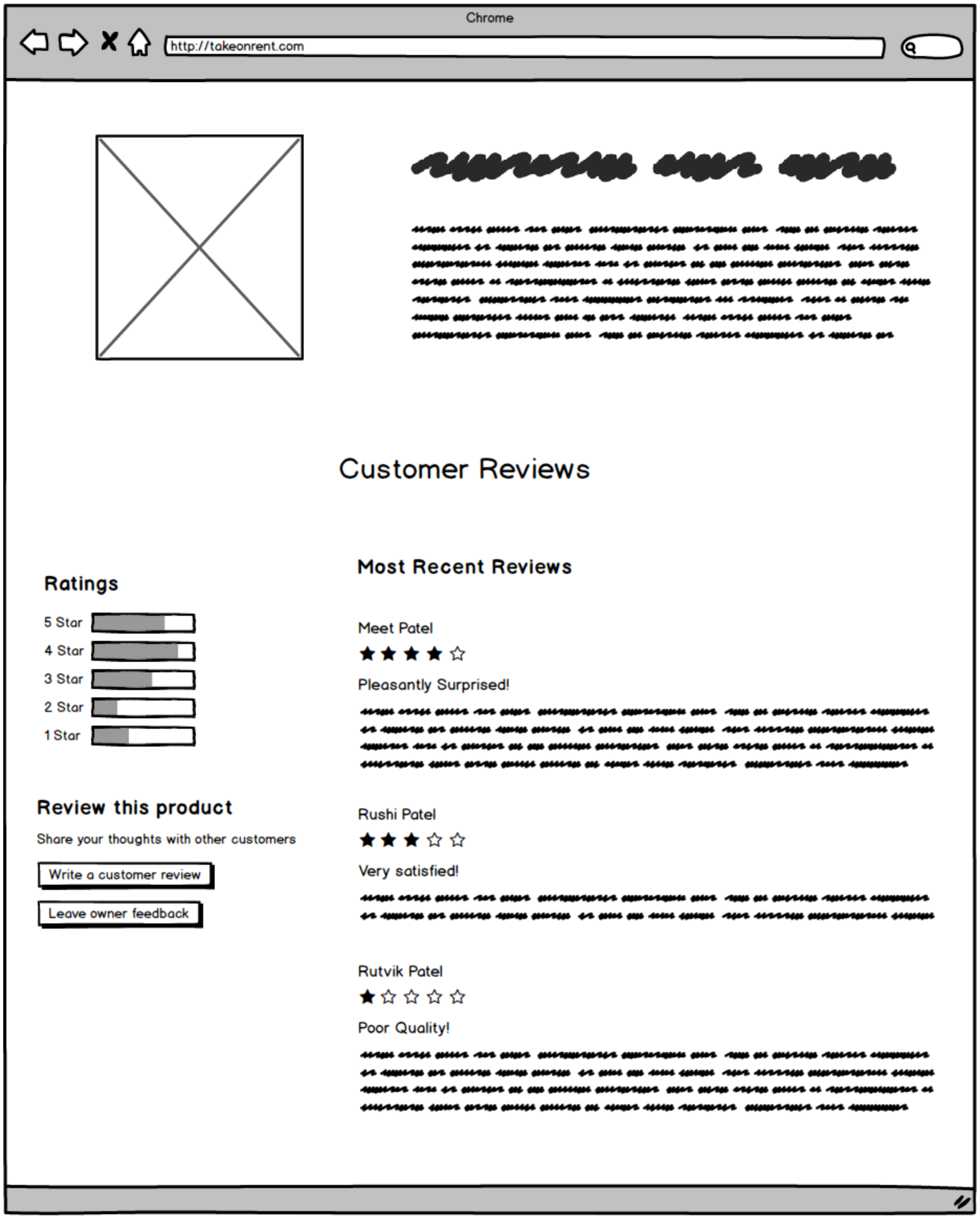


Figure 16: Product page Low Fidelity Template

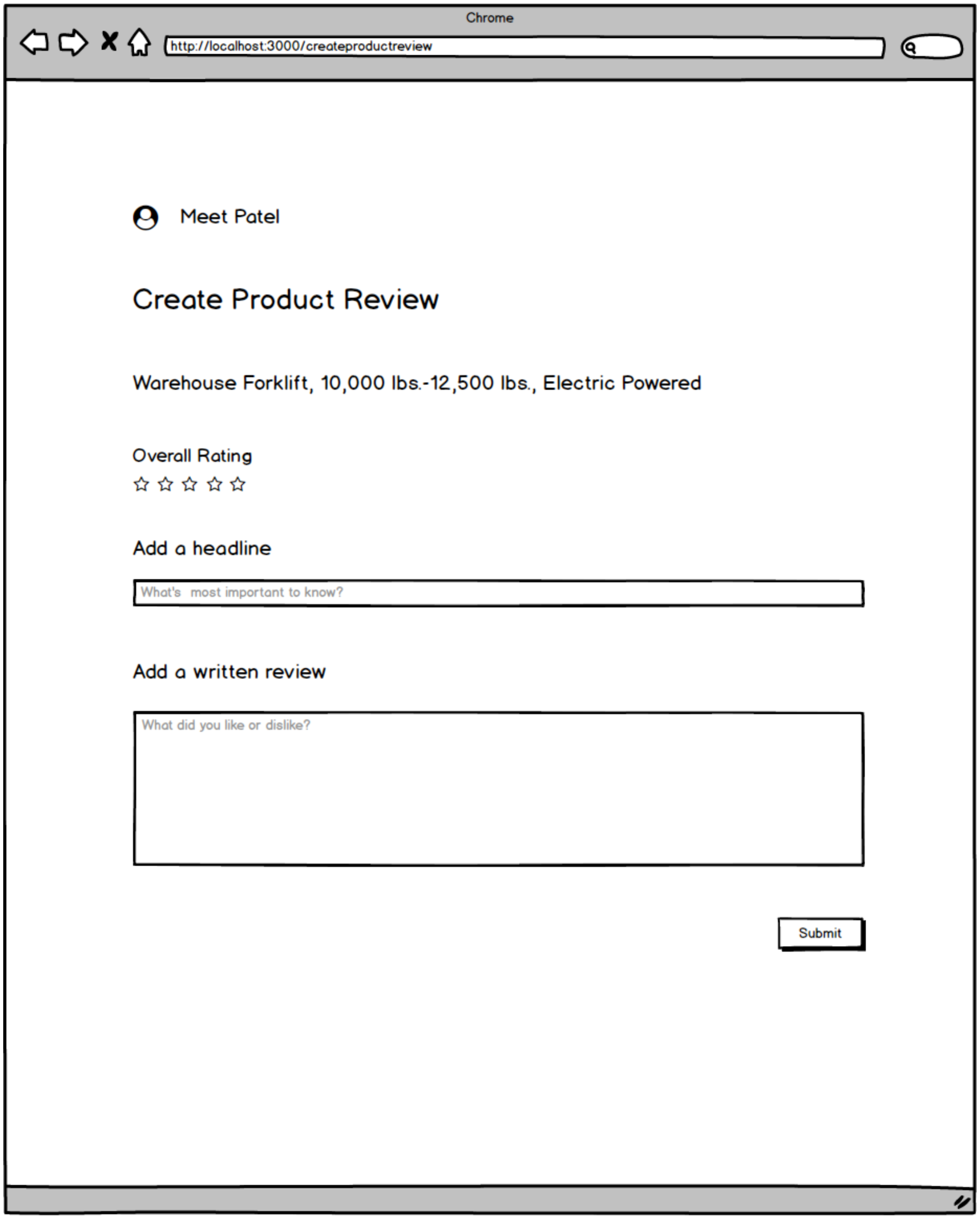


Figure 17: Add Product Review Low Fidelity Template

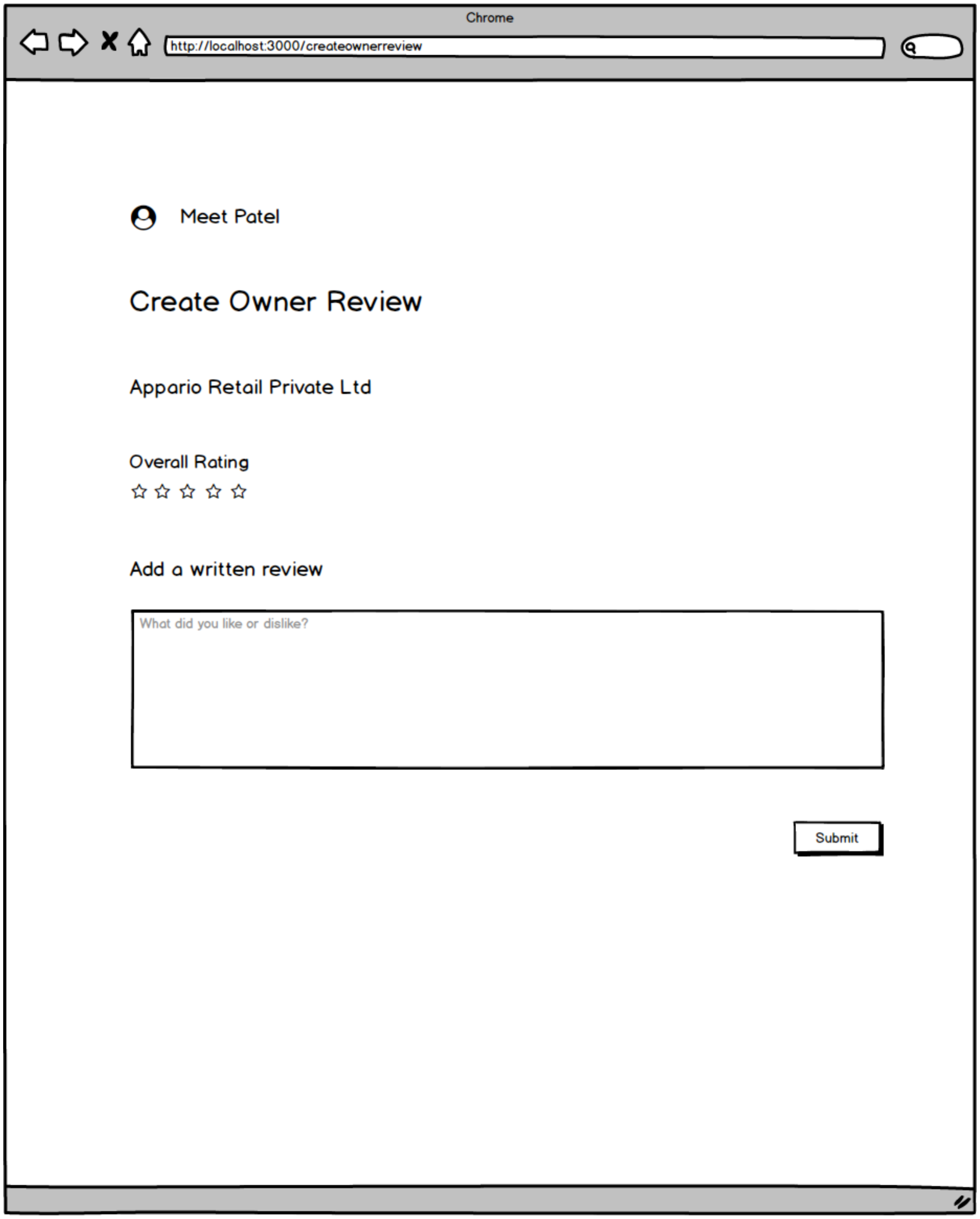


Figure 18: Add Owner Review Low Fidelity Template

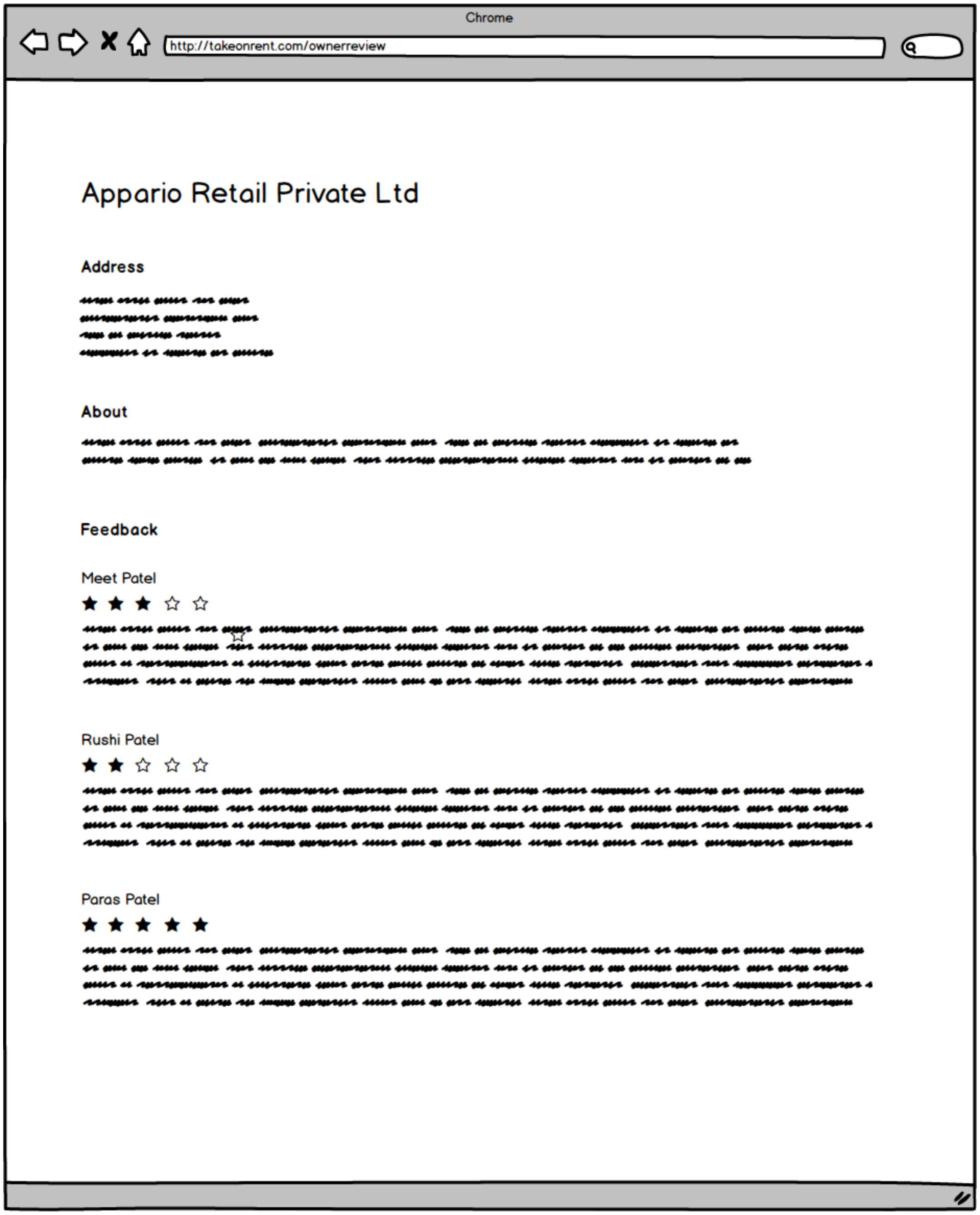


Figure 19: Owner Review Page Low Fidelity Template

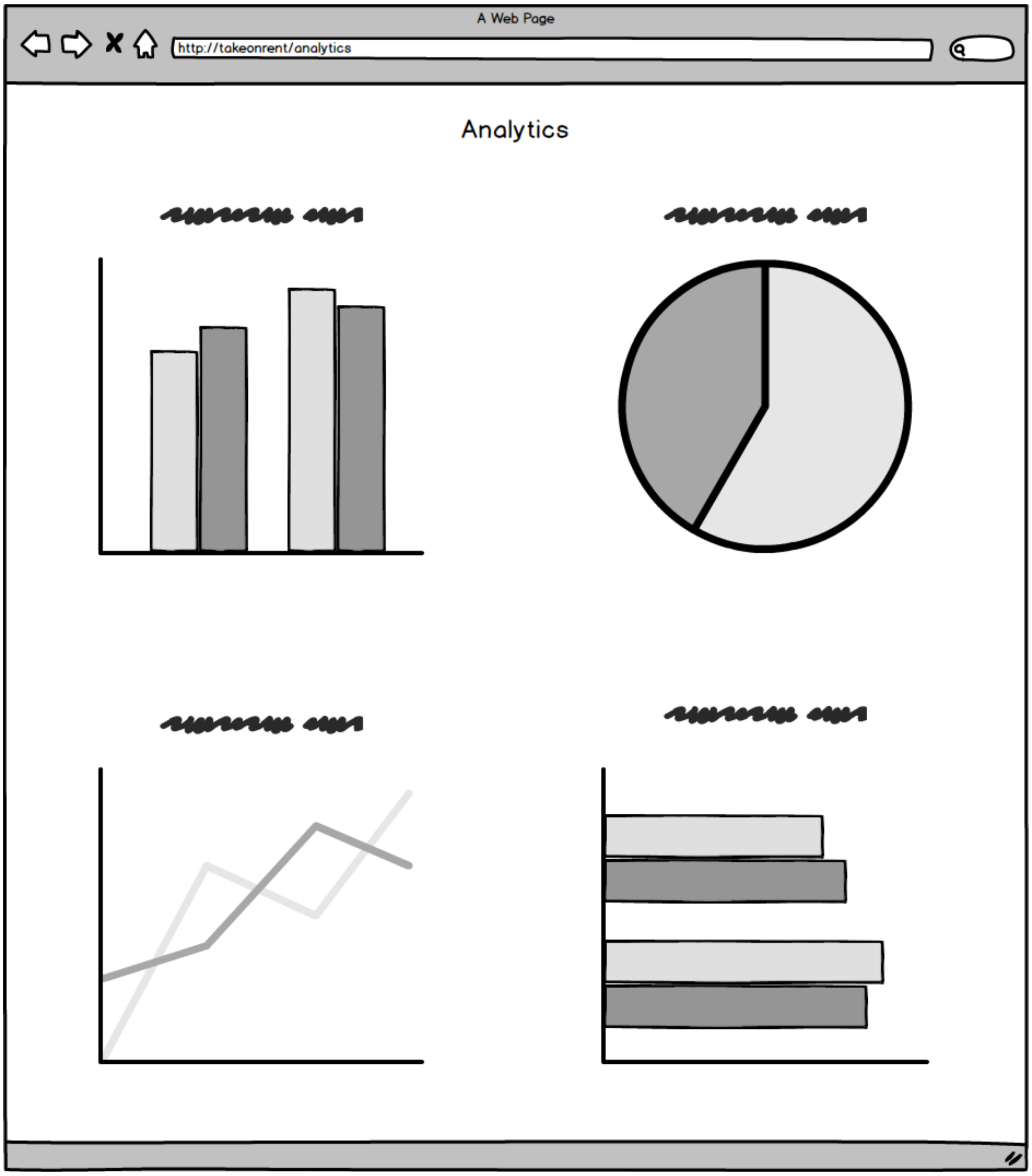


*Figure 20: **Analytics Low Fidelity Template***

Figure 21: Billing Information Low Fidelity Template

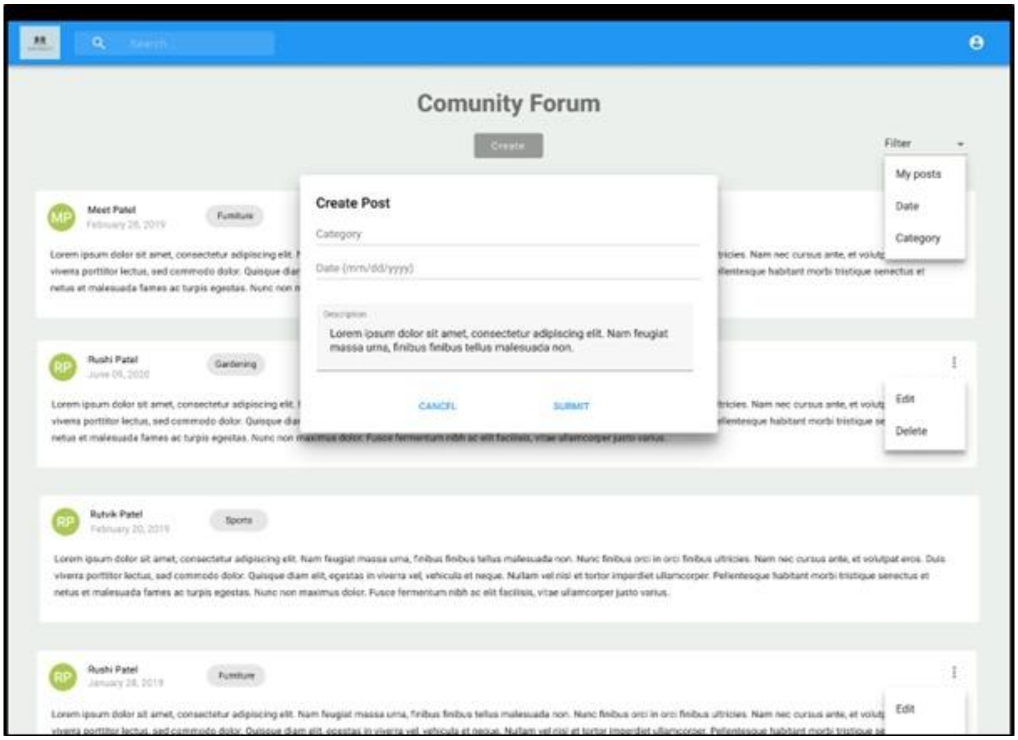


Figure 22 Community forum create post Low Fidelity Template

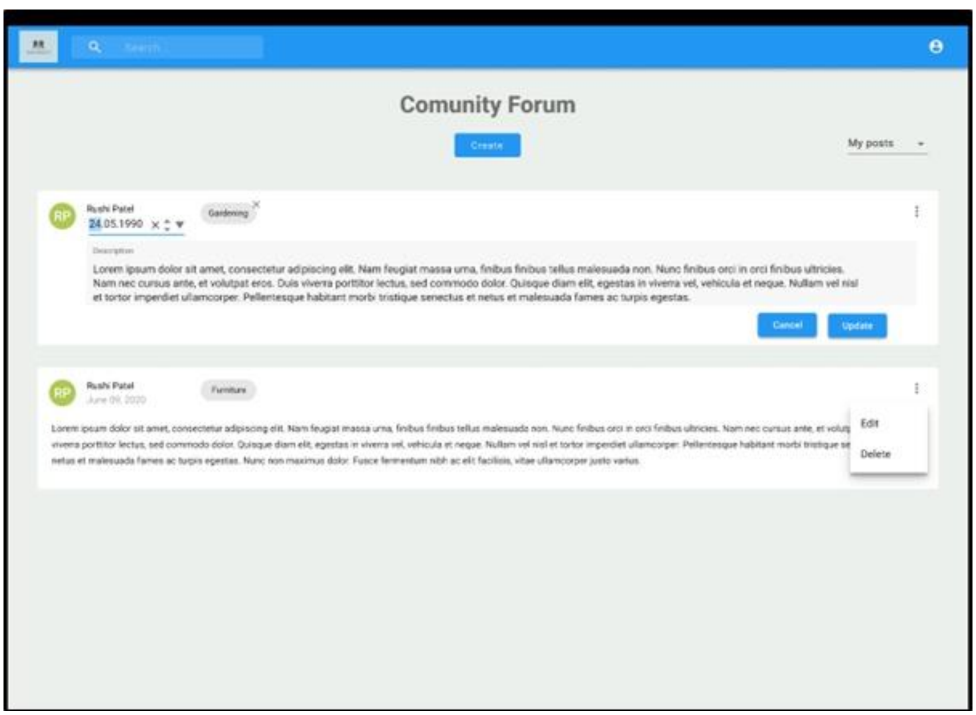


Figure 23: Community forum edit page Low Fidelity Template

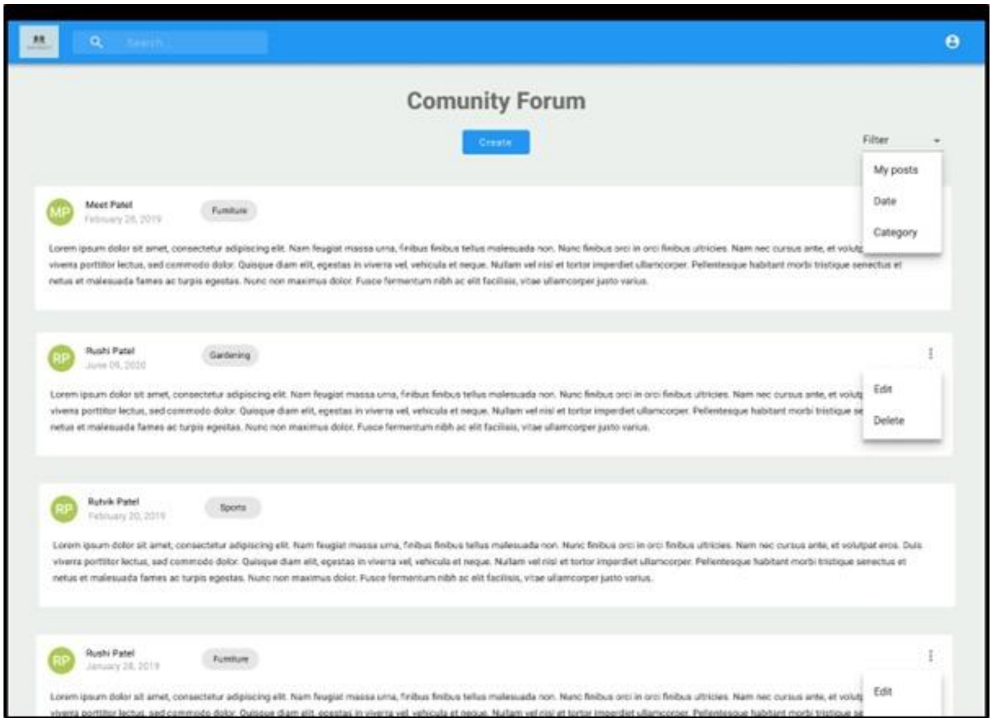


Figure 24: Community forum main page Low Fidelity Template

2.5.3 Website Design

Below are the design decisions for various pages:

While developing the design of the website prime importance has been given to the user's convenience for accessing the website. The entire design of the application has been made responsive using material UI components. Moreover, for ease of access and rectification of errors, informative error messages are shown. The user is now allowed to submit forms until all provided details are correct.

Signup:

For user registration, the sign-up form has been made more interactive and pleasant. The signup form has been shown in the container in which the form and logo are shown side by side. Upon selection of the input field, the input field is focused on the orange colour. It helps the user to identify the location of the field which is currently active.

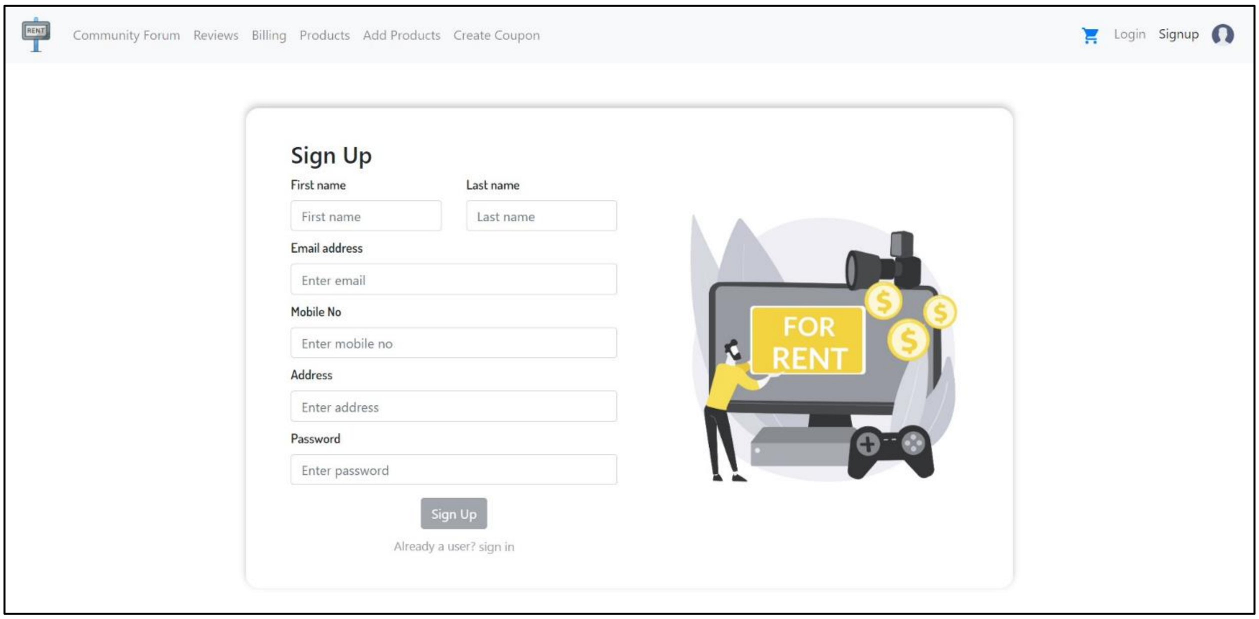


Figure 25: Sign-up page

Sign in:

Sign-in page is designed by keeping in mind the impression of the look and feel of the application. It is not made more spacious instead it has a few fields to accept user details and sections to show informative messages in case of notifying users if incorrect details are provided. The user will not be able to login, until provided all the details are correct.

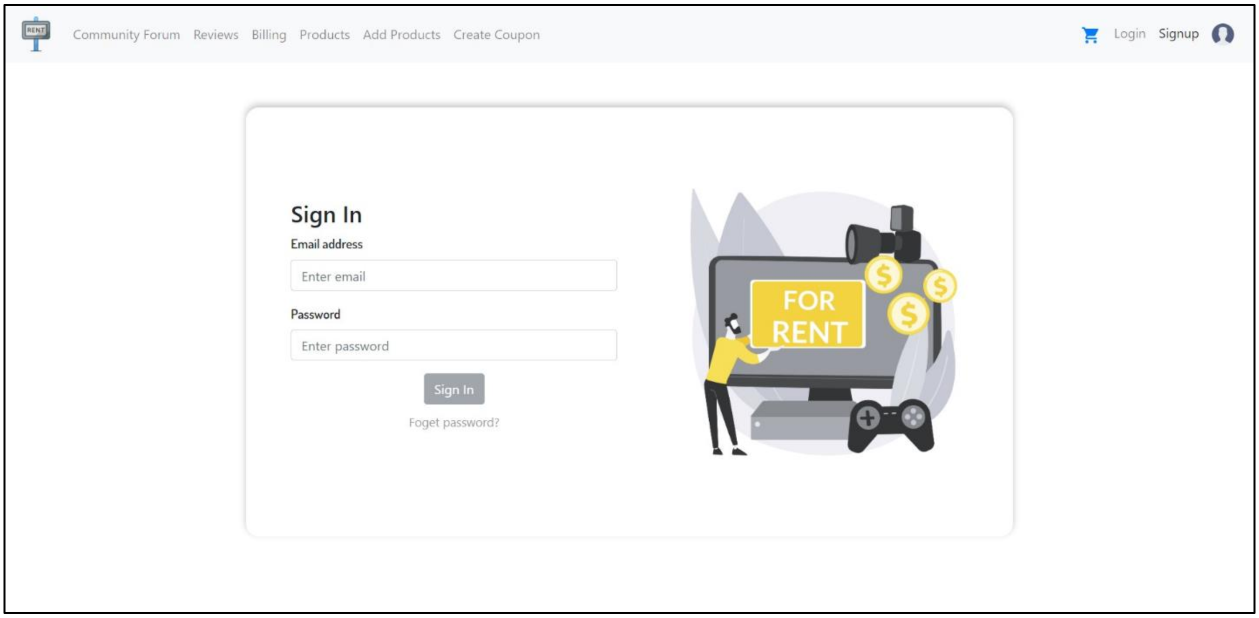


Figure 26: Sign-in Page

Forget Password:

This functionality helps the user to change the password for the account in case the user forgets the password. For this feature, security has been given prime emphasis. It will allow only authorized or registered users to change the password. It will first check whether using the given email id is registered or not. If registered, then the system will send a reset password link on the authorized email id otherwise user will be prompt with the informative message.

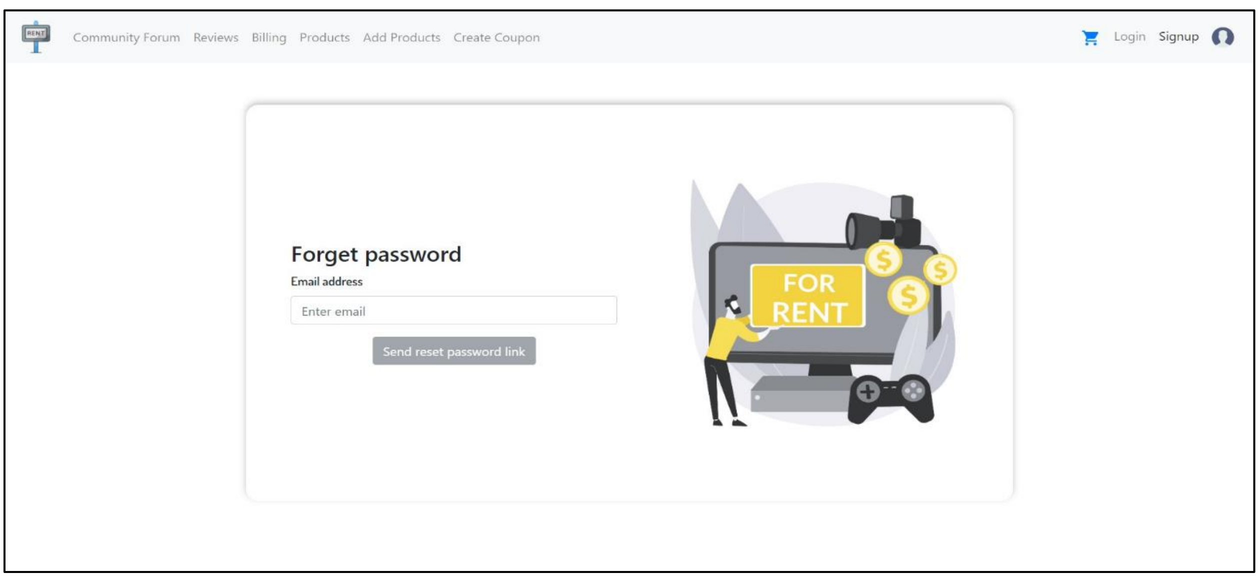


Figure 27: Forget Password Page

Edit Profile:

This page is designed to show user profile details and preferences that the user will update once the user is registered and login into the application. Once the user will provide a preference, the user will get all the products on the product page according to that preference otherwise user will get all the products if the preference is not provided.

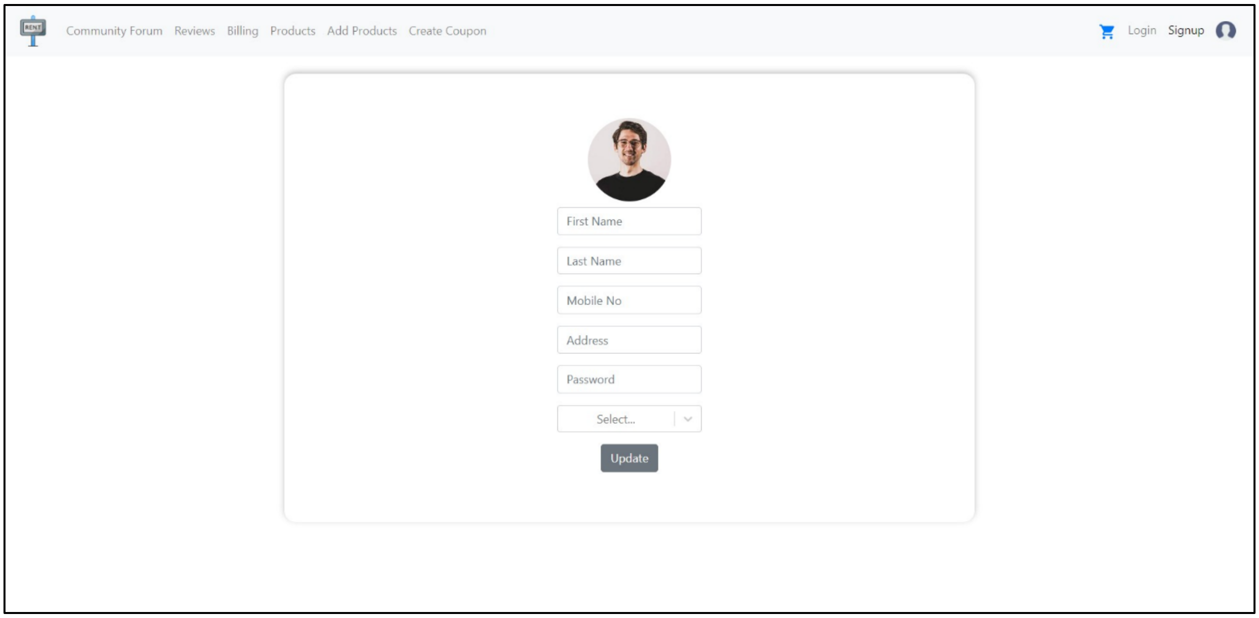


Figure 28: Edit Profile Page

Product catalogue

The product catalog is the page where all the products will be displayed with the required information like product name, product price, product description add the option to add the product to the cart. It also has a filter option to filter the product with respect to the maximum price and name of the products. I have the option to send a message to the owner of the product which pops up in the dialogue box.

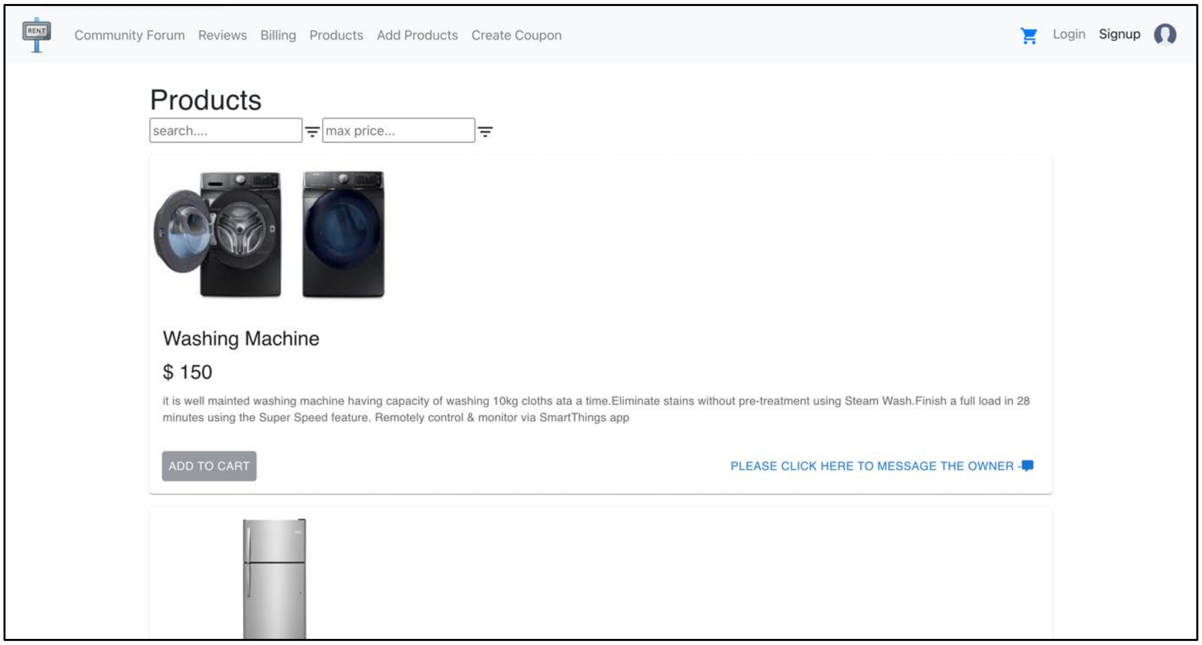
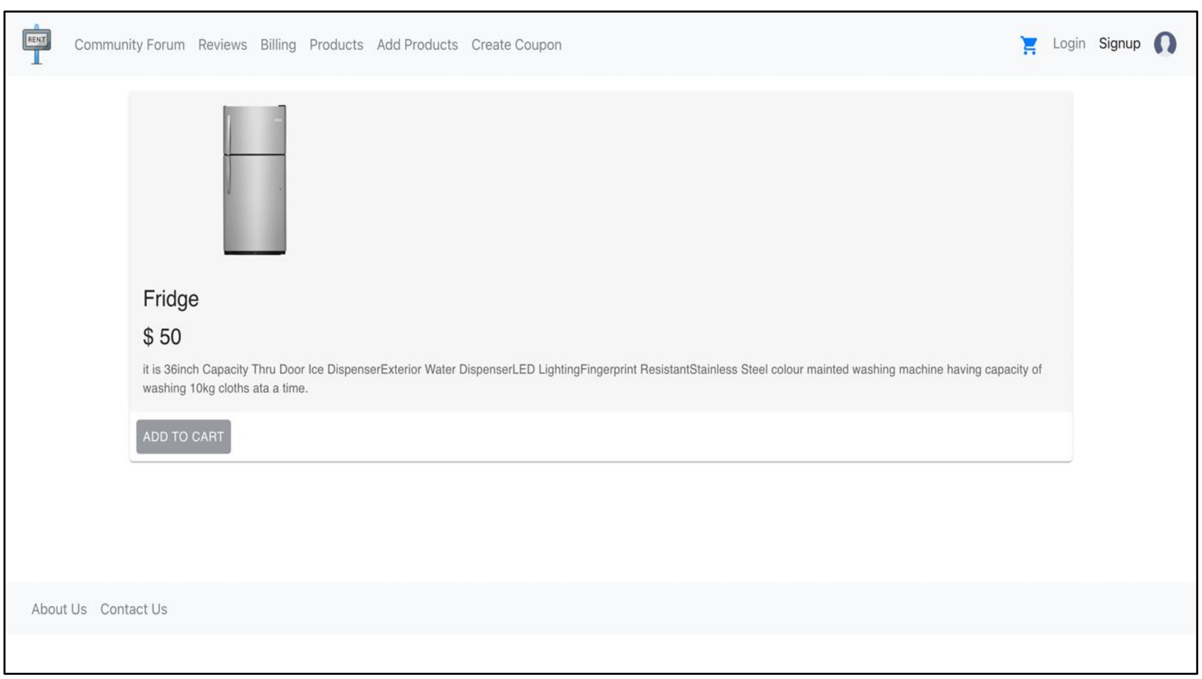


Figure 29: Product Catalogue Page

Single Product Page

The single product is the page where all the detailed information about the single product will be displayed. It has a product name, product price, product description, and option to add a product to the cart.



Empty Cart

The empty cart page will be displayed when there are no products added to the cart.

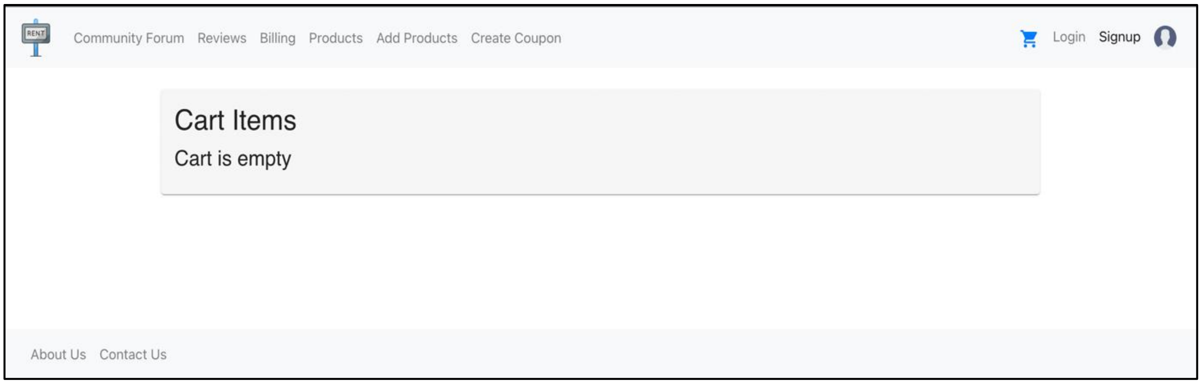


Figure 30: Empty Cart

Cart with products

The cart page will display all the products added to the cart with the option to increase and decrease the quantity. It will also display the total price to pay after adding the tax price and shipping charge to the item price. It has a button to pay the rental amount which will redirect the user to the payment page.

Figure 31: Cart with product

Add Equipment

The **Add Equipment** is the primary page, wherein the owner of the equipment can add equipment of various types. The fields include name, category of equipment, an image of the equipment to be uploaded, description of the equipment, quantity, rent amount, address where the equipment is available, security deposit, for how many days it is available, and lastly, coupon for any discounts to be applied. All the mentioned fields are required and need to be filled in the proper format, except the coupon as it may/may not be applied by the owner. On clicking the ADD PRODUCT button, the product gets successfully added and the owner is redirected to the listings of the product, where it can be seen on the dashboard. Hence, equipment is added and further, it is visible to the customer along with the details. Below attached are the screenshots of the “Add Equipment” page, of which the “Figure-1” indicates the form to be filled by the owner to add the equipment and “Figure-2” indicates the form in a responsive manner.

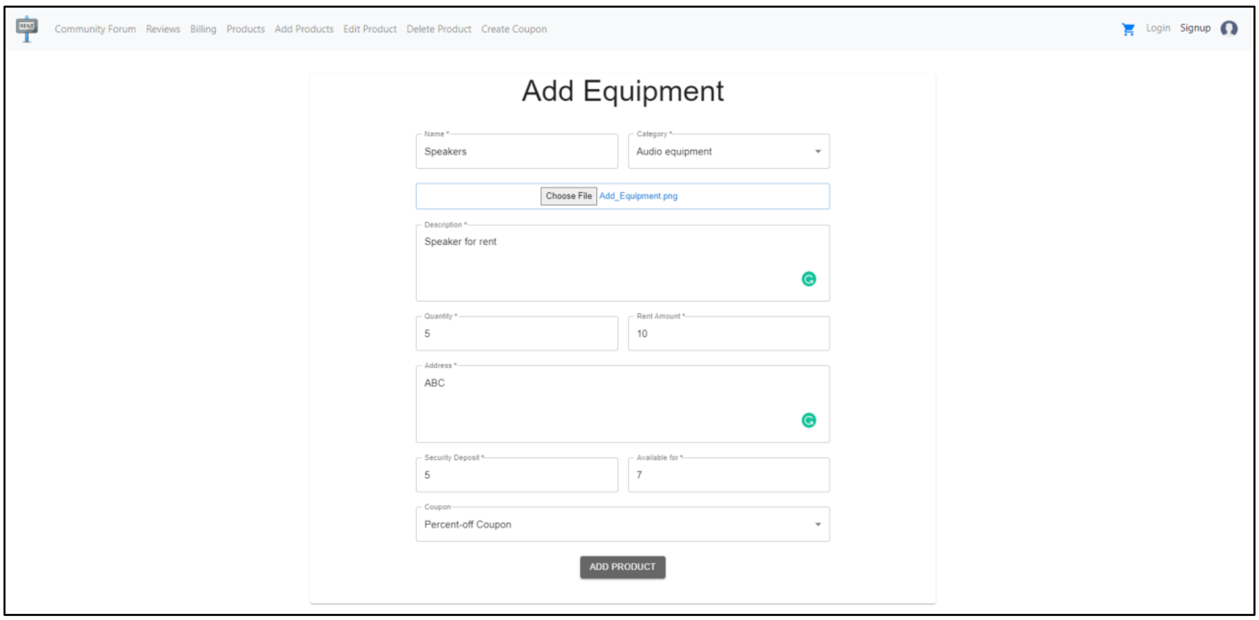


Figure 32: Add Equipment Page

Edit Equipment

The Edit Equipment is the page, wherein the owner of the equipment can edit and update the equipment of various types before publishing it to the dashboard. If the owner wishes to update any of the fields, then the details can be updated on this page. On clicking the SAVE CHANGES button, a dialog appears asking for confirmation from the user for saving the changes (as shown in figure 4), and further, once the button is clicked user is redirected to the dashboard where the product finally gets published. Hence, equipment is added and further, it is visible to the customer along with the details. Below attached are the screenshots of the “Edit Equipment” page, of which the “Figure-3” indicates the filled form that the owner can edit according to the requirements, while “Figure-5” indicates the form in a responsive manner.

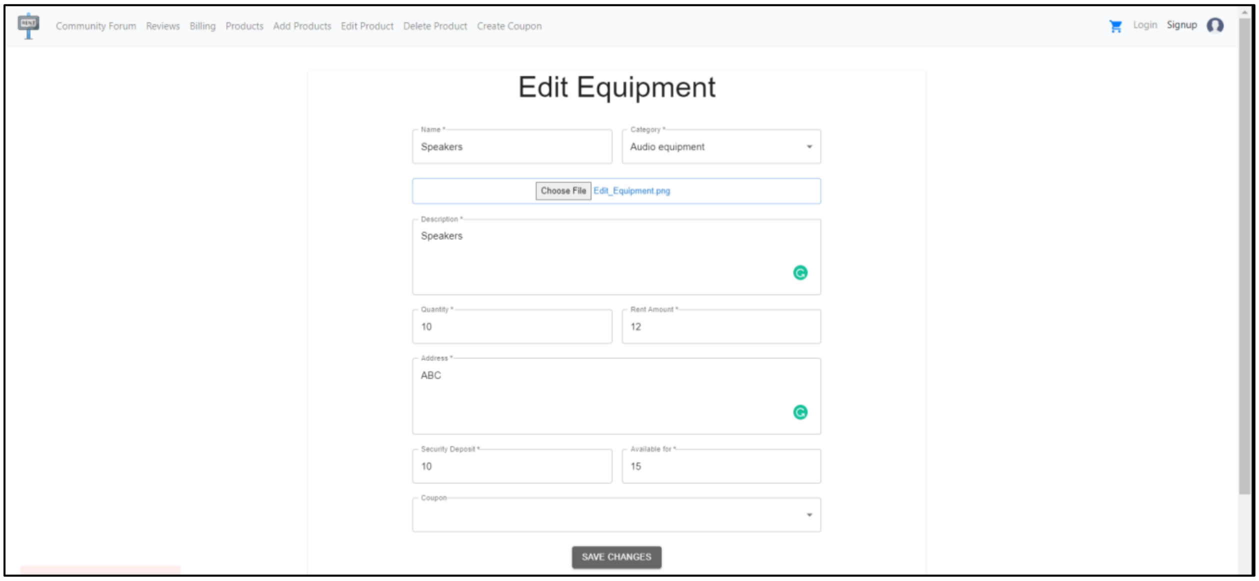


Figure 33: Edit Equipment Page

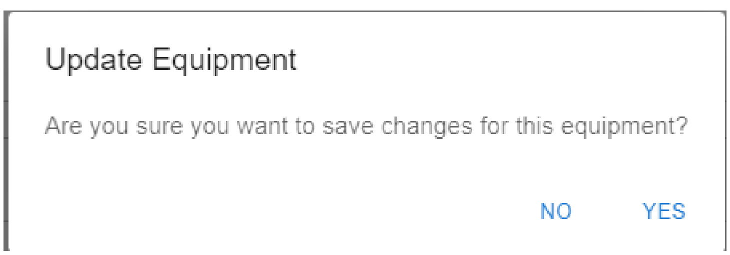


Figure 34: Dialog for confirmation on clicking “SAVE CHANGES” button

Delete Equipment:

The Delete Equipment is the page, wherein the owner of the equipment can view and delete the equipment of various types by reviewing the equipment. On clicking the DELETE button, a dialog appears asking for confirmation from the user for deleting the product (as shown in figure 7), and further, once the button is clicked, the product gets deleted successfully. Hence, equipment is deleted along with its related details and now it is not visible to the customer or owner. Below attached are the screenshots of the “Delete Equipment” page, of which the “Figure-6” indicates the filled form which the owner can delete the equipment, once viewed, while “Figure-8” indicates the form in a responsive manner.

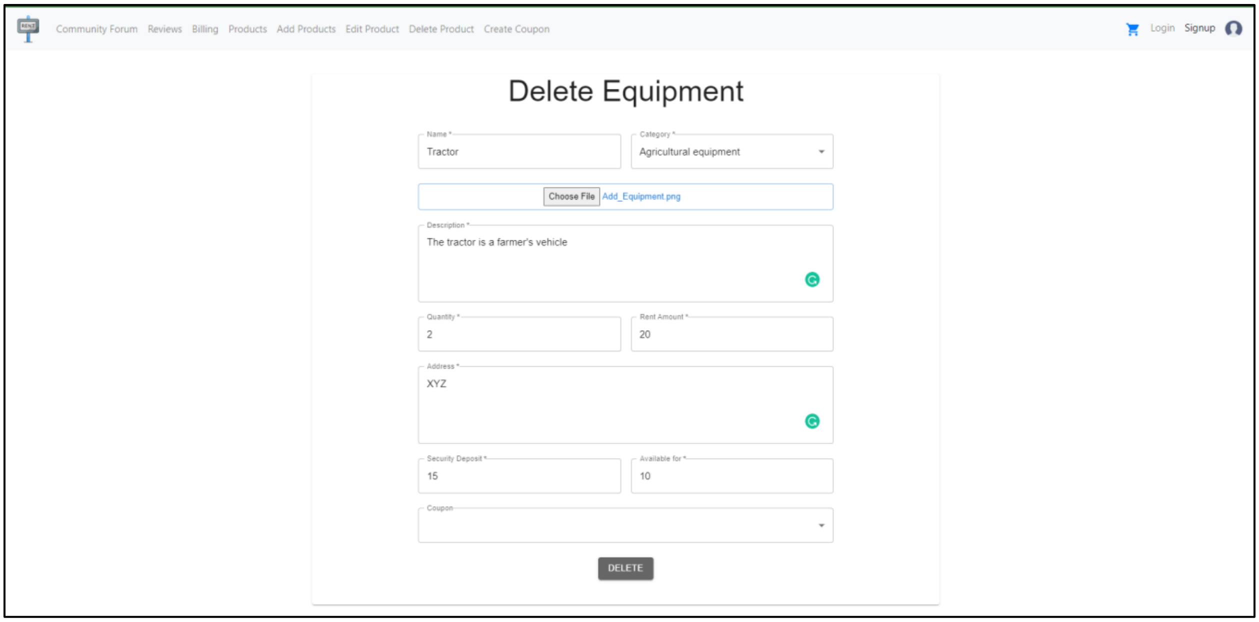


Figure 35: Delete Equipment Page

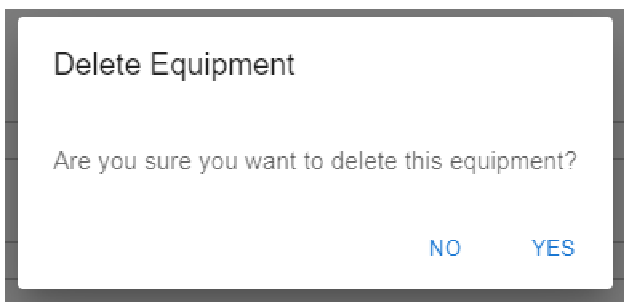


Figure 36: Dialog for confirmation on clicking "DELETE" button

Create Coupon:

The Create Coupon is the page, wherein the owner of the equipment can create and add coupons of various types. The fields include coupon code, which is a unique code assigned to each type of coupon, next to being coupon type, wherein types of coupons such as "BOGO coupon", "Discount coupon", etc., description of the coupon, expiry date of the coupon and lastly, coupon amount, which has the price of many coupons are applied on equipment. All the mentioned fields are required and need to be filled in the proper format. On clicking the ADD COUPON button, the coupon gets successfully added and the owner is redirected to the "Add Product", where it can be added while adding the product by the owner. Hence, equipment is added and can further be applied to any equipment. Below attached are the screenshots of the "Create Coupon" page, of which the "Figure-9" indicates the form to be filled by the owner to create a coupon and "Figure-10" indicates the form in a responsive manner.

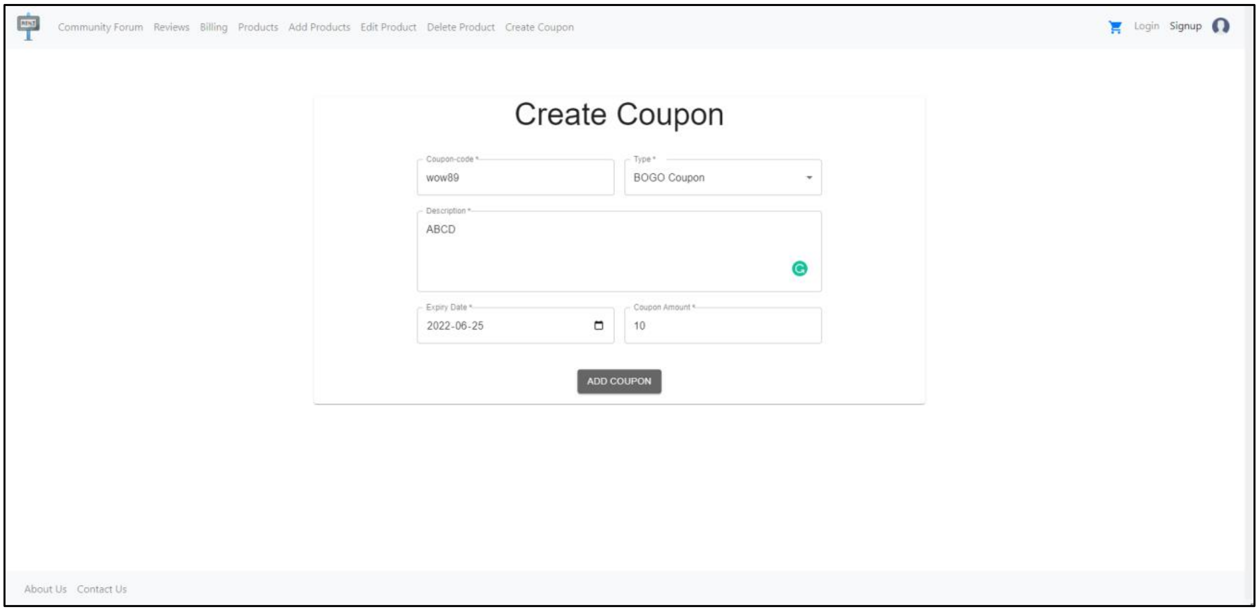


Figure 37: Create Coupon Page

Product Page

The user will be redirected to this page when the user clicks on any product from the product catalogue to get a detailed description. On this product page, the user will get a detailed description of the product along with a button to add that product to the cart. As the user scrolls down the user will be able to see the customer reviews of the product along with an overall rating of the product. Currently, there are four buttons One is the "Write a Customer Review" button to write the customer review of the product. Another is the "Leave a seller feedback" button to rate the product owners. Next, there is an "All Posted review" button to see all the posted reviews by the user. Finally, there is an analytic button to open the analytics page.

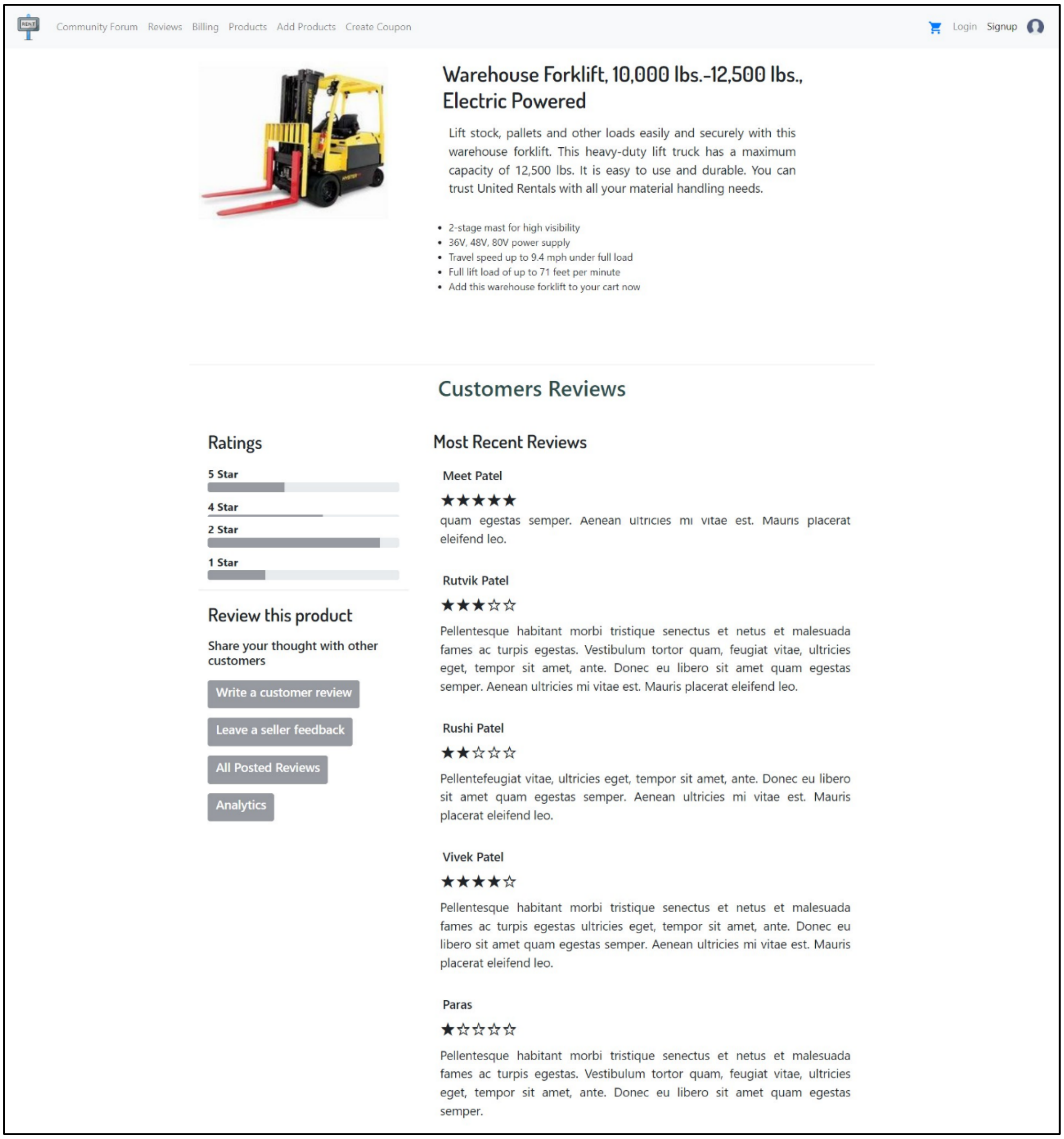


Figure 38: Product Page

Create Product Review

This page is responsible for posting product reviews. On this page, users need to provide ratings for the product, the headline, and the written review of the product. On submitting this information, the review of the product will be posted. In the written review the user needs to write about what the user likes or dislikes about the product and need to share his overall experience with the product.

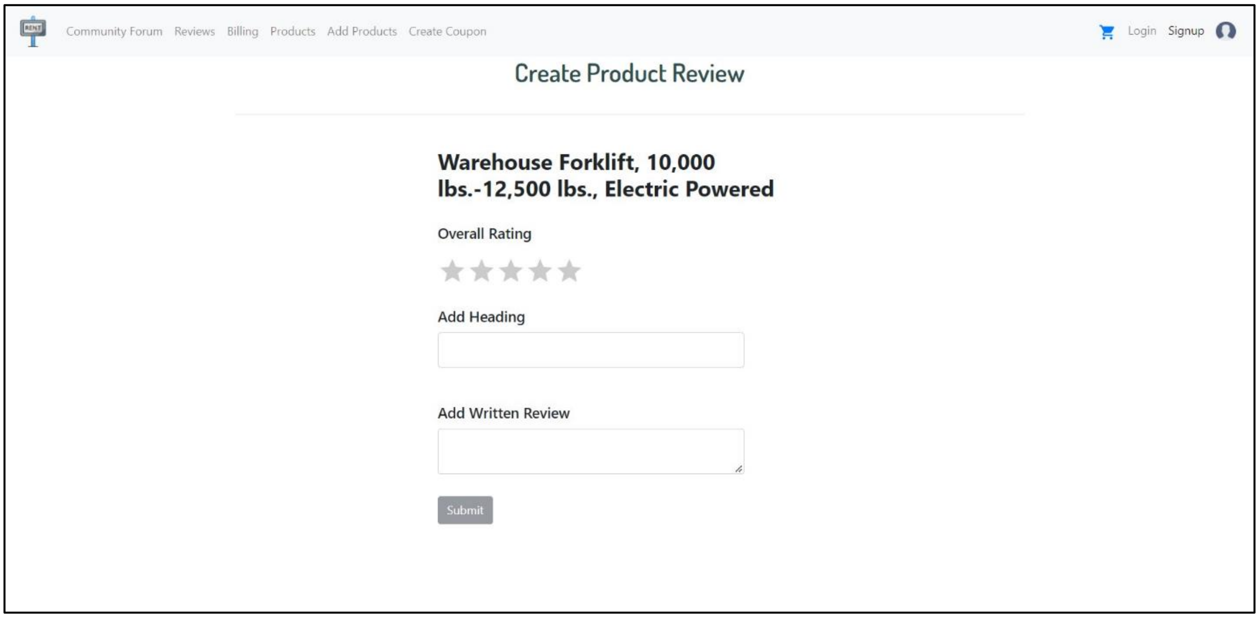


Figure 39: Create Product Review Page

Create Owner Review

This page is responsible for posting product owner reviews. On this page, users need to provide ratings, the headline, and the written review of the product owner. On submitting this information, the review of the product owner will be posted. In the written review the user needs to share his overall experience with the product owner from renting a product till the user returns it.

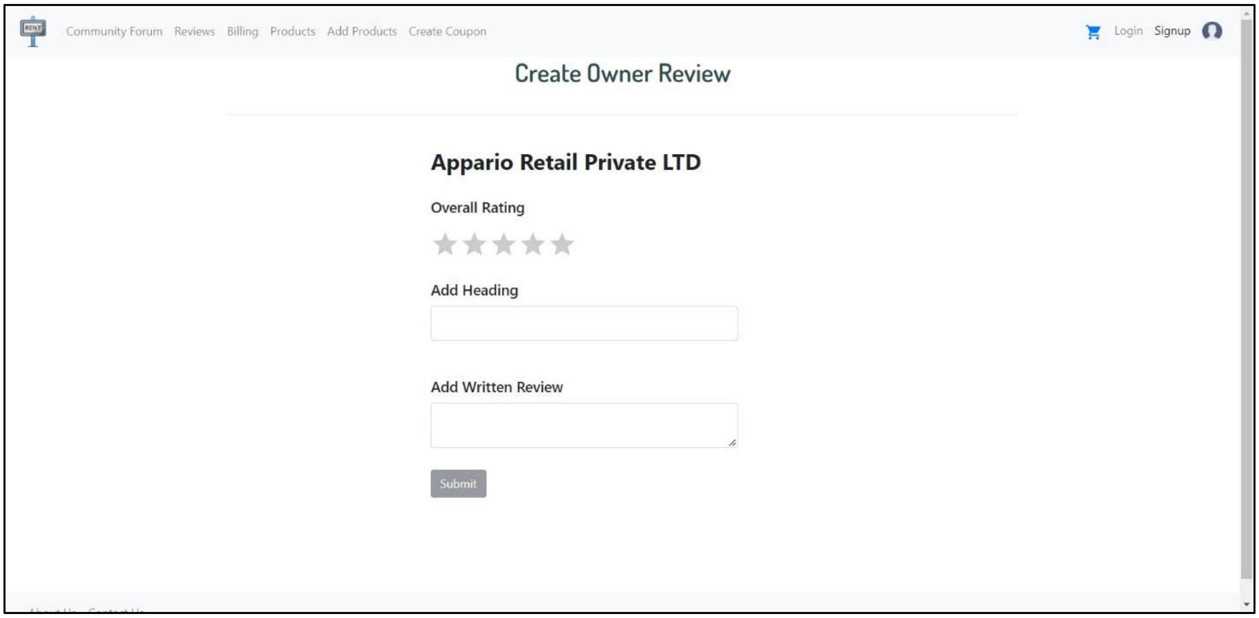


Figure 40: Create Owner Review Page

Posted Reviews

This page is responsible for displaying all the reviews posted by an individual. On this page, the user can edit as well as delete any review.

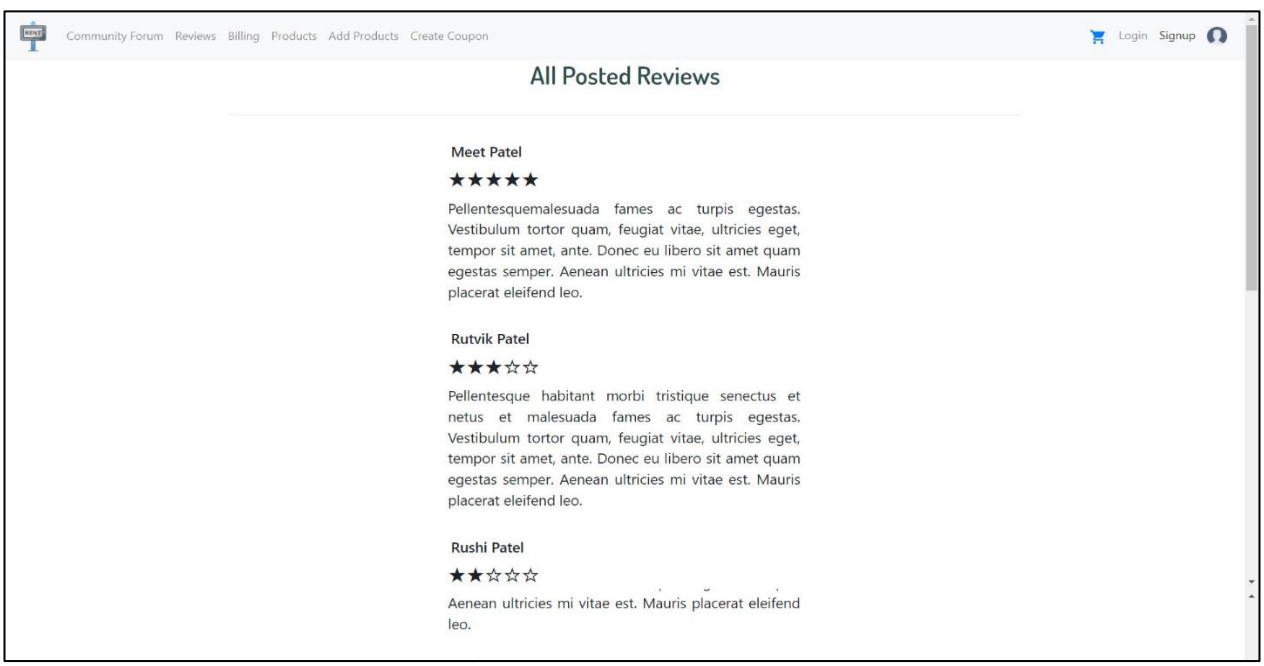


Figure 41: Posted Review Page

Analytics Page

The analytic page is responsible for representing various analyses done on the "TakeOnRent" platform. For example, it shows a chart of the number of users vs months, products rented vs months and much more.

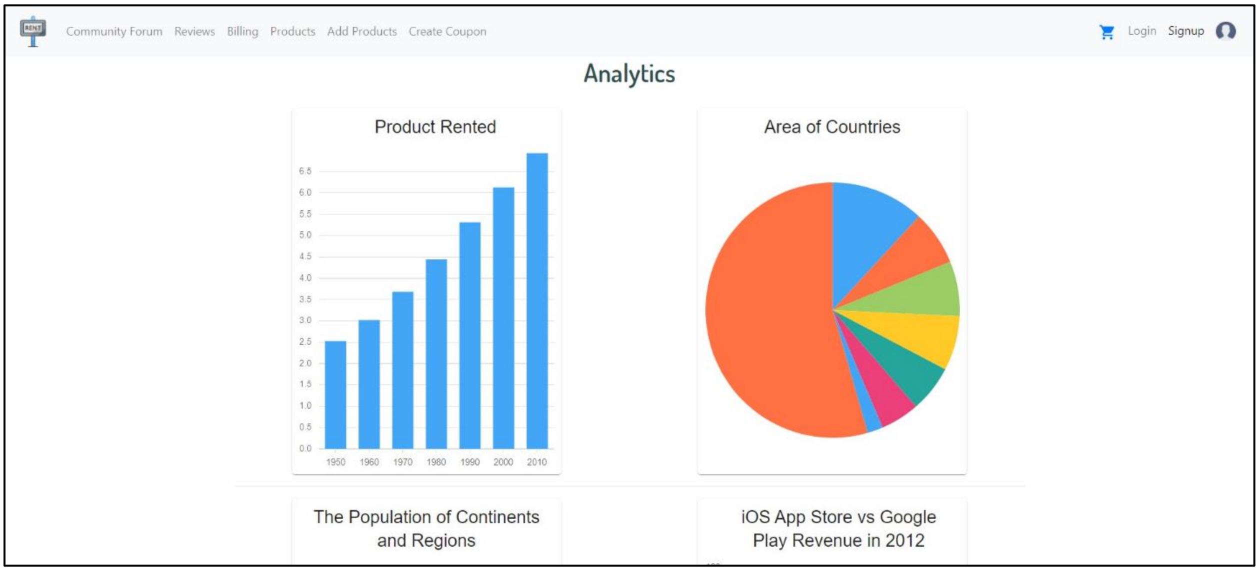


Figure 42: Analytic Page

Billing and Insurance Page

On this billing page, we have four sections. First, we have a saved address section where if the user is our returning customer, he can directly pick the address from it or add a new one. Then we have asked for the personal information of the user, such as First, Last Name, Phone Number, and email address. Phone number and email address are validated and need to be 10 digit and valid email format. Then if the user chooses to add a new address, the address information section needs to be filled; it will be disabled and automatically select the user saved address.

At last, we have a piece of payment information that will show the summary of the order and the total amount, along with giving an option to the user to select the insurance coverage for their rental equipment. And then, it will provide a choice of payment, including MasterCard, Visa, PayPal, etc. When they submit the order, it will give them a payment confirmation message.

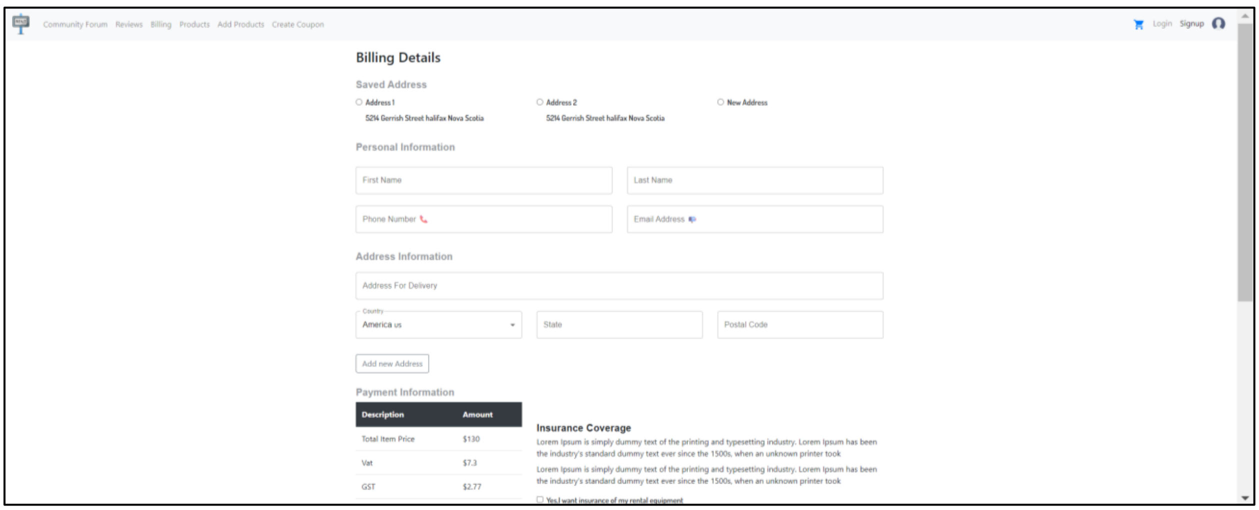


Figure 43: Billing Page

Community Forum:

The main page for the community forum was build using Material UI library. The design of this page is completely responsive and is user friendly. The purpose of this page is for the users or the owners to post regarding a specific product. The owner can post regarding a product he wants to put on rent for better reach while the users can view all the posts, filter based on their interested category and find the best and updated status of the product that they would like to rent. The features of this page include creating a post, updating a post, deleting a post, viewing all posts publicly and filtering posts based on category, date or 'my posts'. The screenshots of the UI of this feature are displayed below:

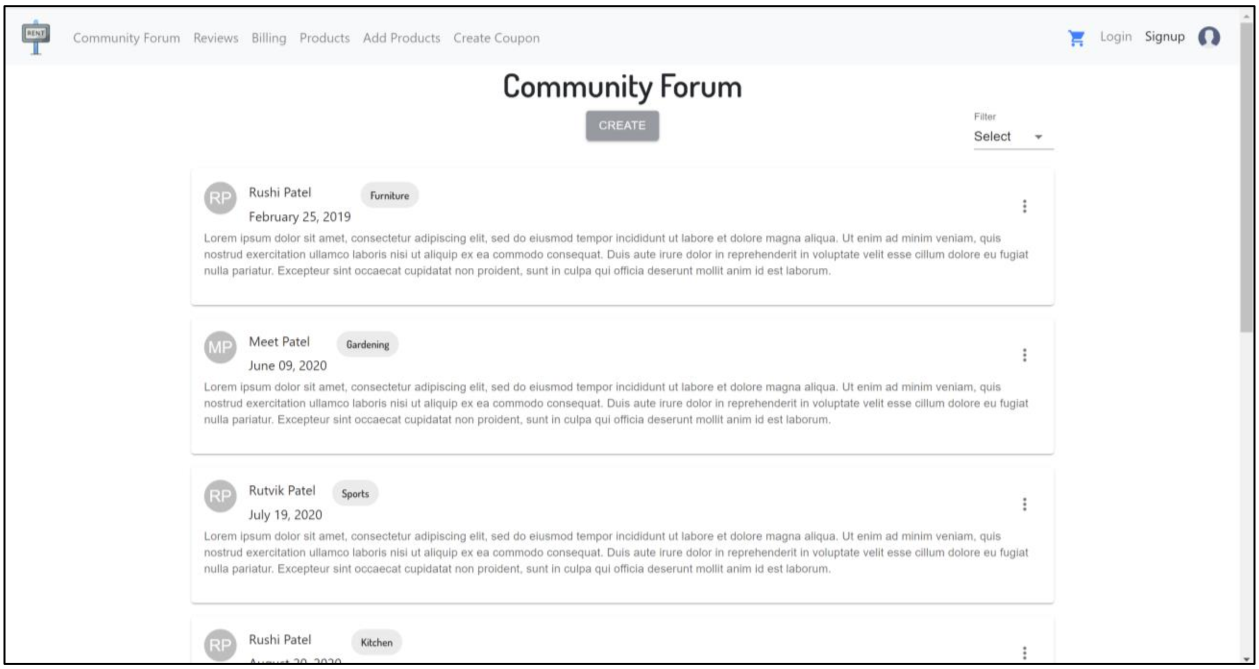


Figure 44: Community forum main page

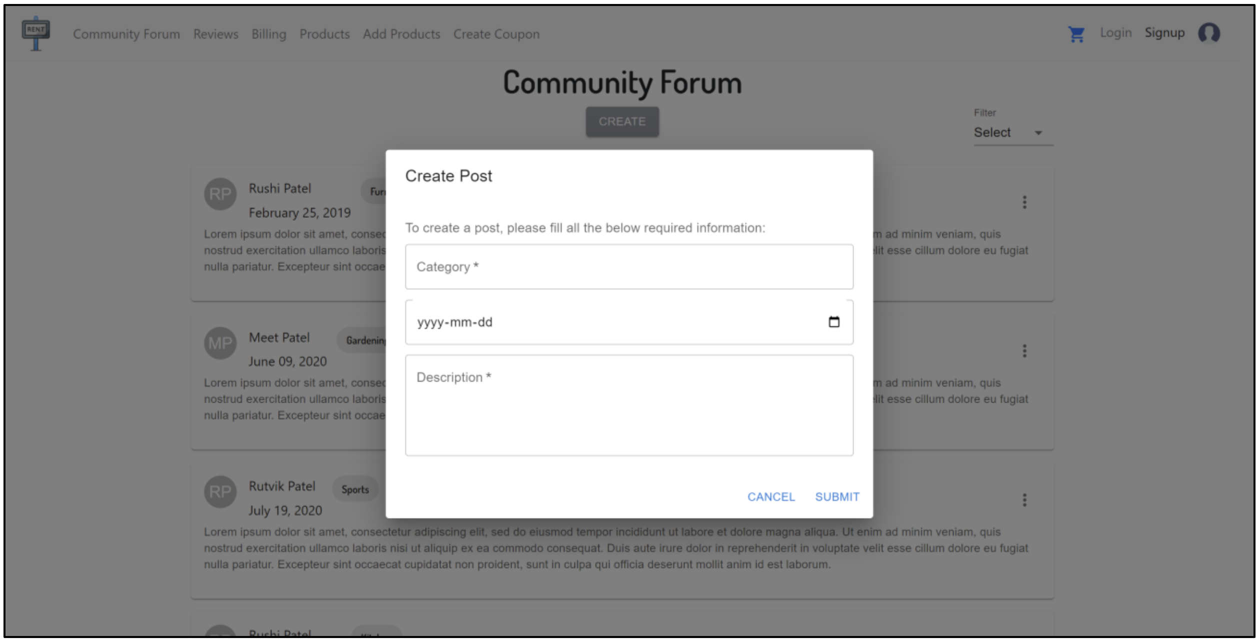


Figure 45: Create post in Community forum

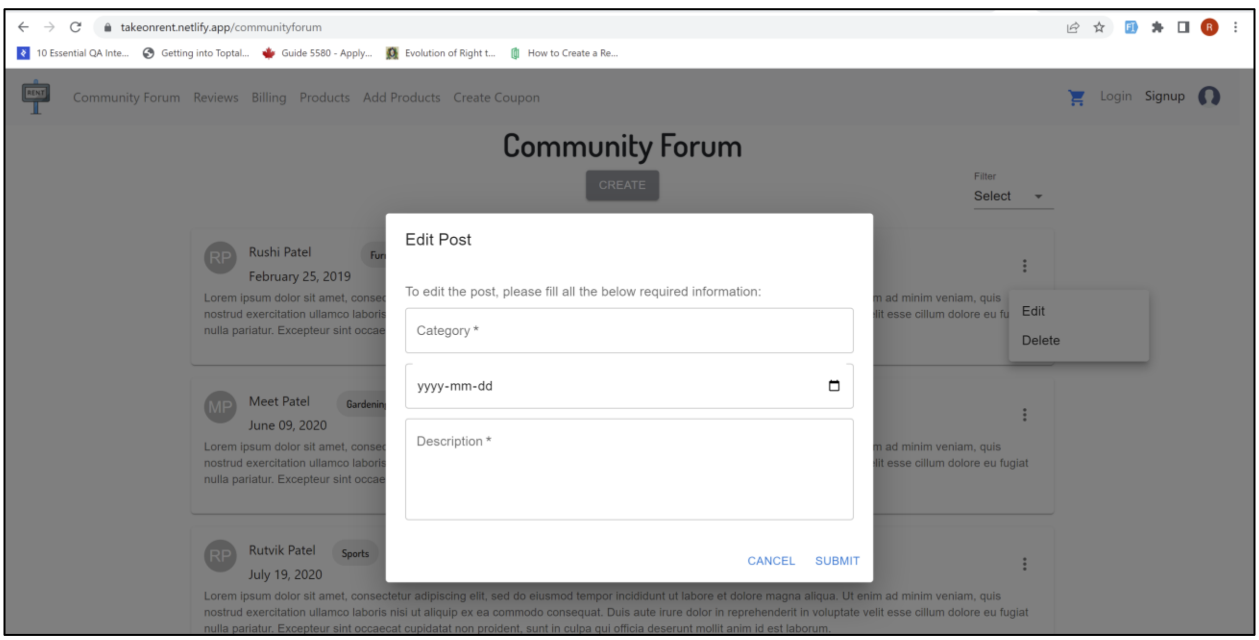


Figure 46: Edit post in Community forum

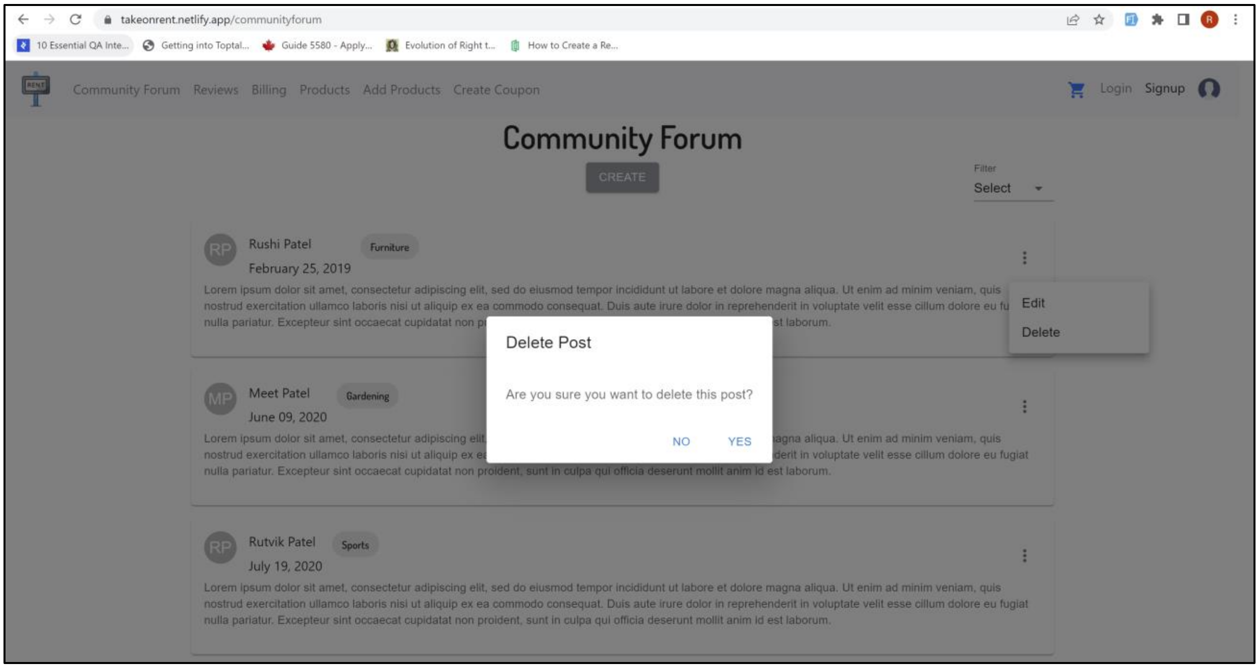


Figure 47: Delete post in Community forum

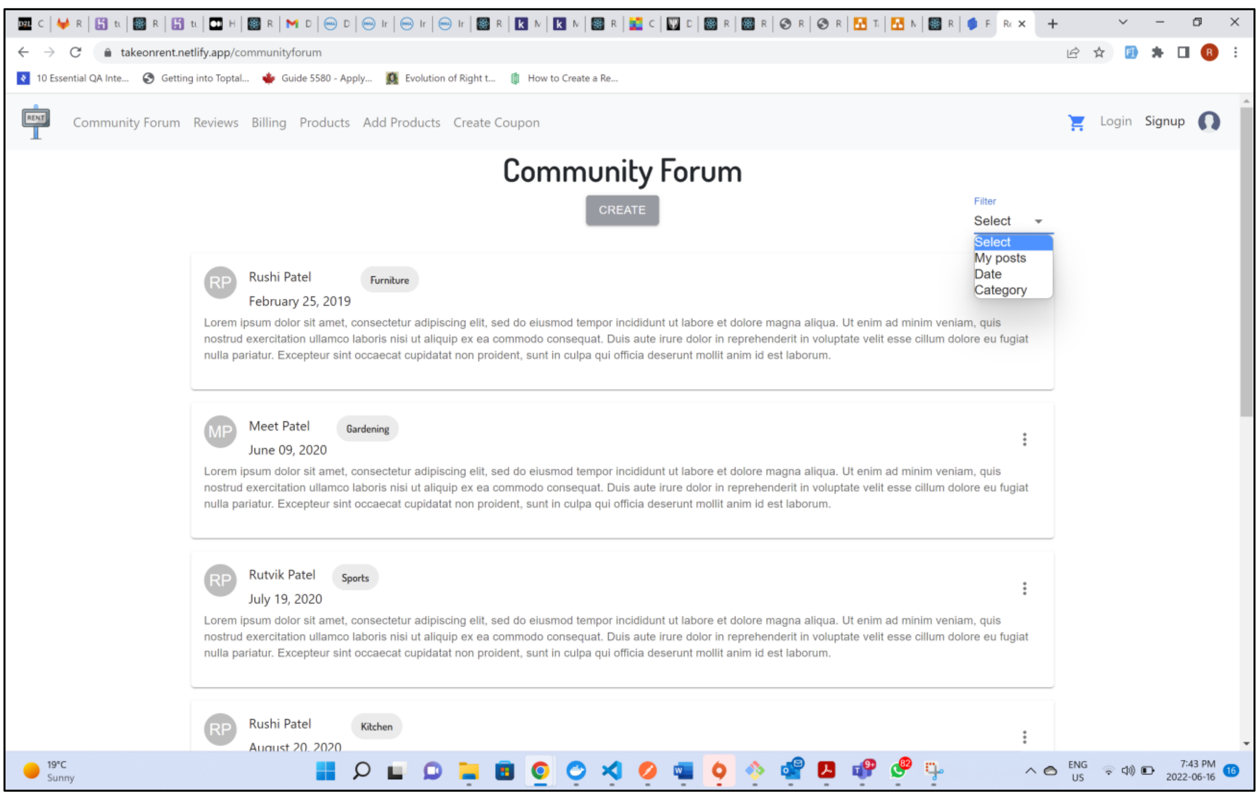


Figure 48: Apply Filter on community Forum

Messaging Service

The messaging service feature is designed using Material UI library and has a completely responsive design. The design of this feature is integrated with the product display feature. The reason of integrating it with the products page was to make the application user friendly and the flexibility to conveniently message the owner of the product while he is viewing the product. The button to message the owner is placed besides the add to cart button. On clicking the messaging button, a form will be displayed to the user having required fields of subject and the description of the message he wants to send to the owner. On successfully submitting the form, the message would be sent to the owner's registered email address from the users registered email address. The screenshots of this feature are displayed below.

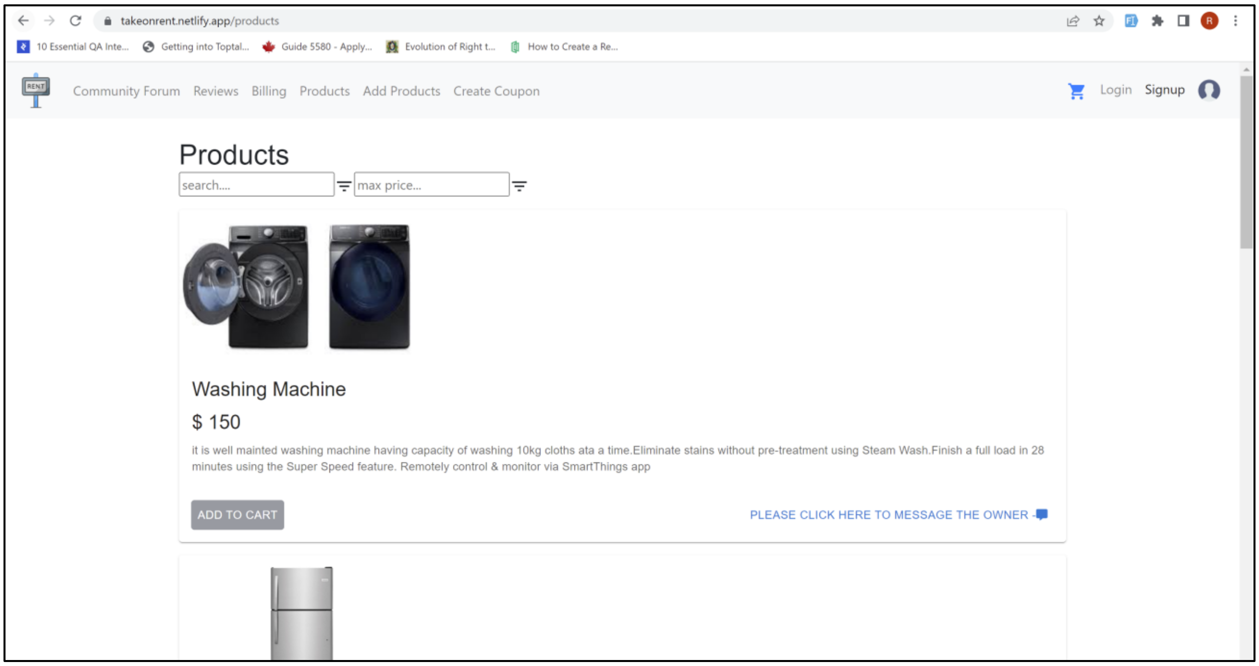


Figure 49: Messaging feature integrated with products

2.6 User Experience

1. User management

Task 1: Signup

Scenario: This feature will be used, when the new user creates an account to put any useful equipment on rent.

- **User Persona:** The user who wants to put equipment they own on rent and earn some extra money.
- **Need:** To allow each user to utilize the equipment they own to earn an extra amount by giving them on rent instead of keeping them idle and getting dust on it.
- **Context:** A user owns equipment such as a driller, table, wheelbarrows, and garden cart. All this equipment is not in active use. Thus, the user wants to put this equipment on rent and make some extra income instead of keeping all equipment idle.

Use case:

1. User is at the Signup page of the TakeOnRent (https://TakeOnRent.herokuapp.com/signup).
2. User clicks on the Signup button at the top right corner of the page.
3. User is redirected to the Signup page.
4. The user gets a form asking for certain details such as first name, last name, email, address, and password.
5. User fills in all the details
    a. User gets error messages in red color if mandatory details are not provided.
    b. User gets an error message in red color if any details are not added in the required format.
    c. User gets an error message if the user tried to register with an email address by which account is already created.
6. Once the user enters all the correct details submit button gets enabled.
7. User submit the button.
8. User gets an account verification email on the given email address.
9. User clicks on an account verification link.
    a. If the user is unable to verify the account after clicking on the link user, click again on the link and verify the account.
10. Once the account is verified successfully then the user account is activated and the user is redirected to the login page of the application(https://TakeOnRent.herokuapp.com/signin).

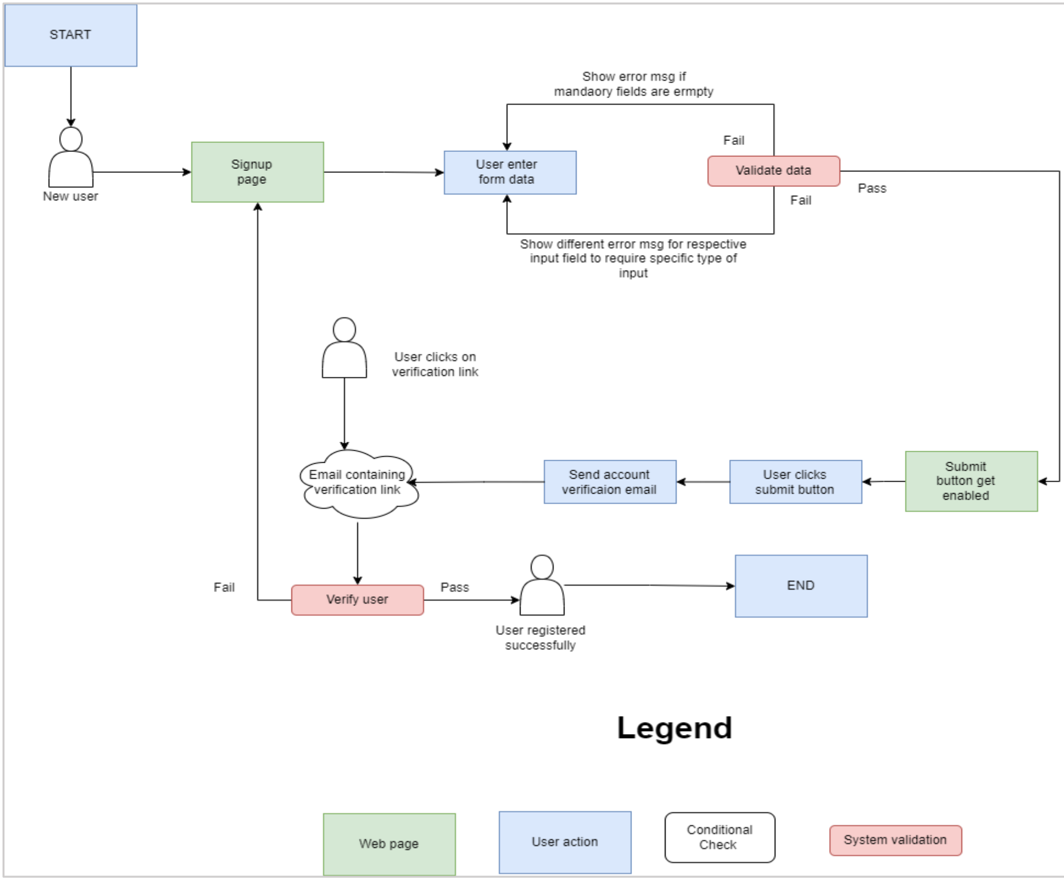


Figure 50: User signup task flow

Task 2: Sign-in

Scenario: This feature will be used when the user has already created an account on TakeOnRent.

- **User Persona:** The user who also thinks that instead of keeping their equipment idle they can be put rent on to make some extra income
- **Need:** For the optimum utilization of equipment user own.
- **Context:** A user who wants to put equipment on rent, login into his account and add details for the equipment and make equipment available for taking on rent.
-

Use case:

1. User is at the Sign-in page of the TakeOnRent (https://TakeOnRent.herokuapp.com/signin).
2. User clicks on the Login button at the top right corner of the page.

3. User is redirected to the Sign-in page.
4. The user gets a form asking for credentials such as email address and password.
5. User fills in all the details
    a. User gets error messages in red color if mandatory details are not provided.
    b. User gets an error message in red color if the user adds incorrect credentials.
6. Once the user enters all details correctly submit button gets enabled.
7. Once the user submits the button, the user is authenticated, and the user is redirected to the product catalog page. Where users can search for various equipment to get them on rent.

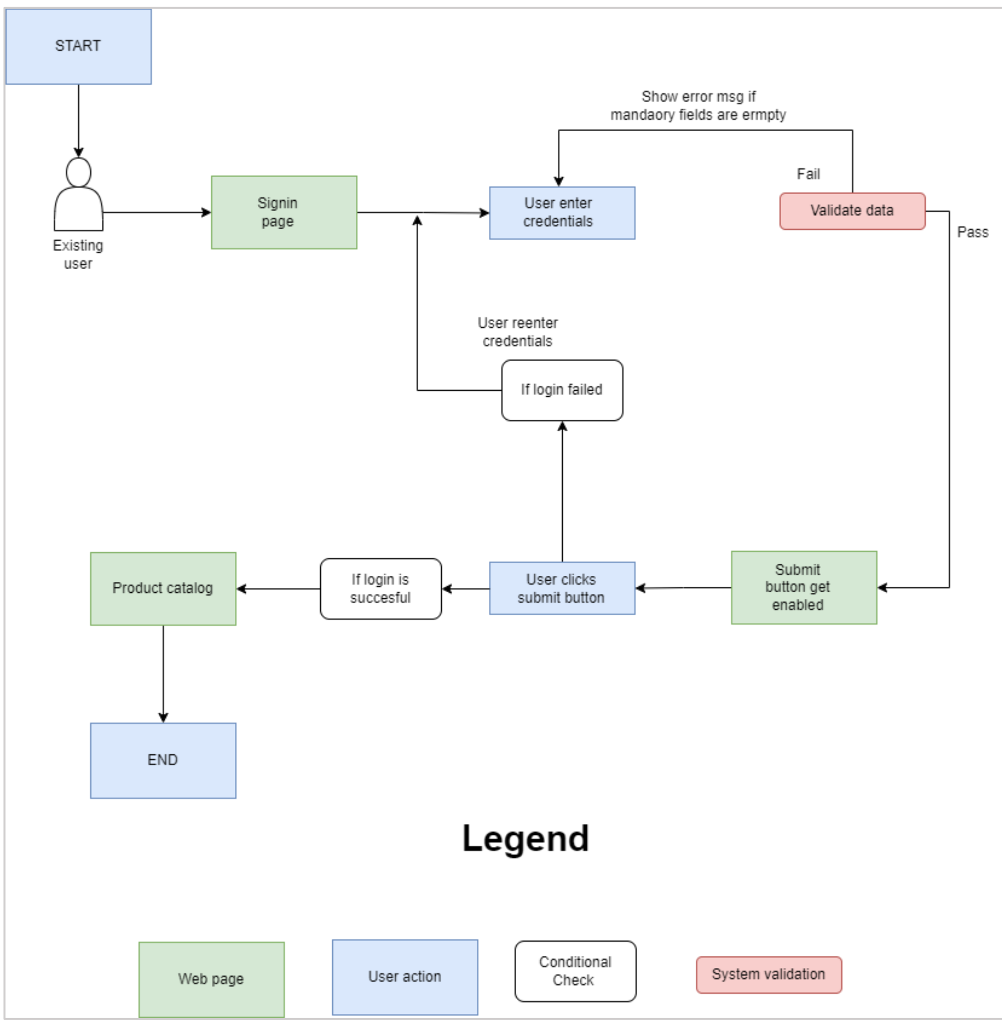


Figure 51: User Sign in Task Flow

Task 3: Forget Password

Scenario: This feature will be used, when users don't remember the password of their account.

- **User Persona:** The user wants to put their equipment on rent but is unable to remember the password they set for the account.
- **Need:** To reset the password for the account, that which user doesn't remember. The user tried multiple times but was unable to recall the password.
- **Context:** The user visits the website to put equipment on rent but the user is unable to log in as the user does not remember the password the user set when the account was created.

Use case:

1. User is at the Forget password page of the TakeOnRent (https://TakeOnRent.herokuapp.com/forgetpswd).
2. User clicks on the Forget password link below the submit button of the login page.
3. User is redirected to the Forget password page.
4. The user gets a form asking for the registered email address.
5. User fills in all the details
   a. User gets error messages in red colour if mandatory details are not provided.
   b. User gets an error message in red colour if the user does not provide a registered email address.
6. Once the user enters all details correctly Send verification link button gets enabled.
7. Once the user submits that button, the user gets an email containing a reset password link on the registered email address.

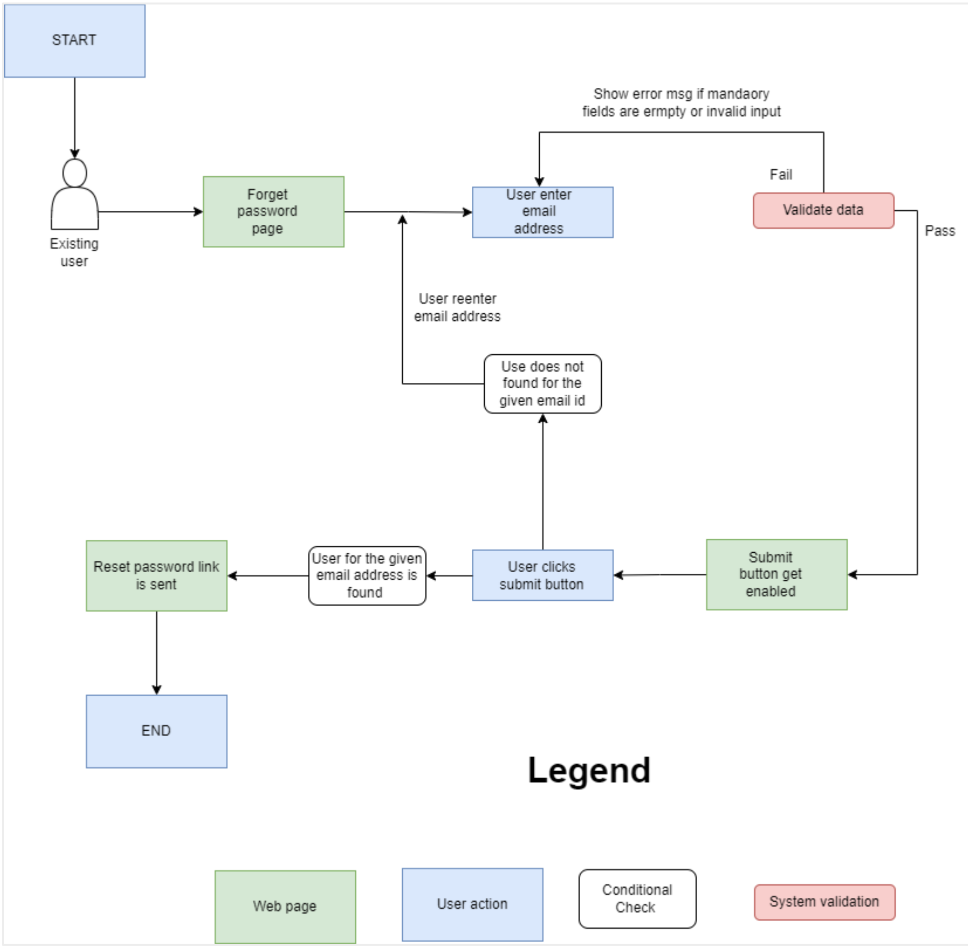


Figure 13: Forget Password Task Flow

2. User Profile

Scenario: This feature will be when user wants to change their profile details such as first name, last name, email, mobile no and password.

- **User Persona**: The user wants to change their profile details which they can use for further actions such as getting an email notification and showing it to other users.
- **Need: To reset** the password for the account, that which user doesn't remember. As well as change profile details.
- **Context**: The user visits the website to put equipment on rent but the user wants to change profile details which are visible to other public users.

Use case:

8. User is at the Forget password page of the TakeOnRent (https://TakeOnRent.herokuapp.com/forgetpswd).
9. User clicks on the Forget password link below the submit button of the login page.
10. User is redirected to the Forget password page.
11. The user gets a form asking for the registered email address.
12. User fills in all the details
    a. User gets error messages in red colour if mandatory details are not provided.
    b. User gets an error message in red colour if the user does not provide a registered email address.
13. Once the user enters all details correctly Send verification link button gets enabled.
14. Once the user submits that button, the user gets an email containing a reset password link on the registered email address.

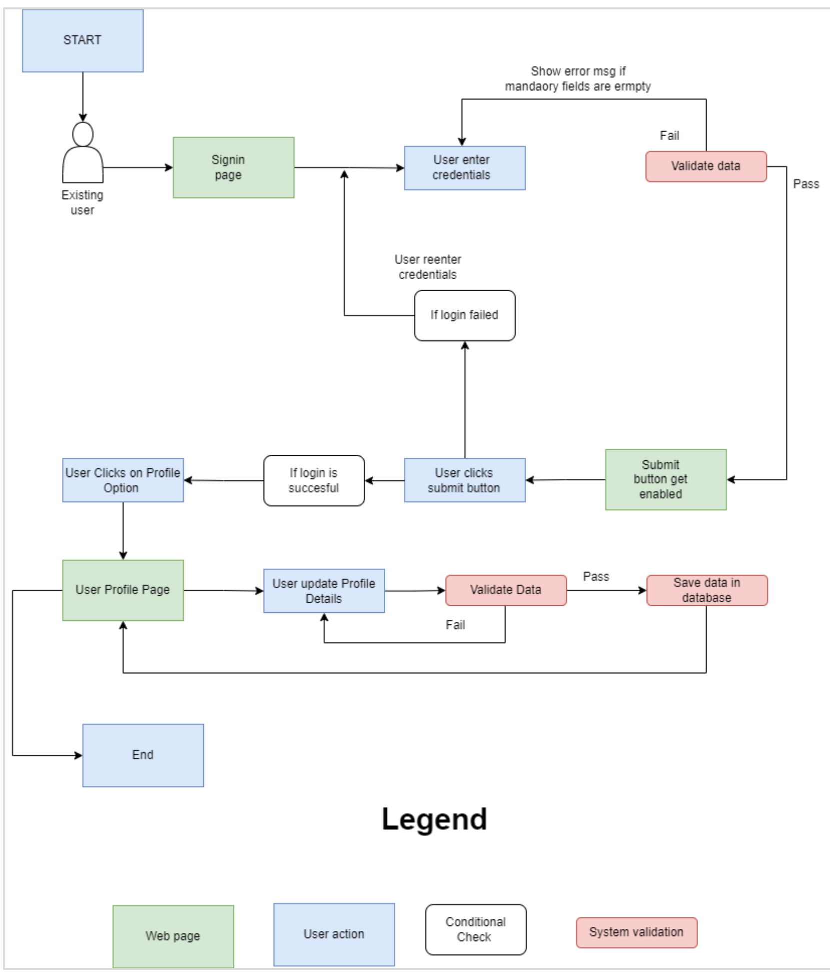


Figure 52: User Profile Task Flow

3. User Reviews

Task 1: Add Product Review

Scenario: The user recently rented one of the products from the **TakeOnRent** website. Now the user wants to share his experience with others so that they get an idea about the product.

Use Case:

Assume that the user is a registered user and has rented one of the products from the website

1. Users open the browser and visit "TakeOnRent.com"
2. User signs into his account
3. User then opens the product that he already rented
4. In the customer review section, the user clicks on the button which is labelled as "Write A customer review"
5. A new page opens where the user gives an overall rating, enters the title, and writes the product review
6. User then submits the review
7. Finally, the review will be posted on the product page.

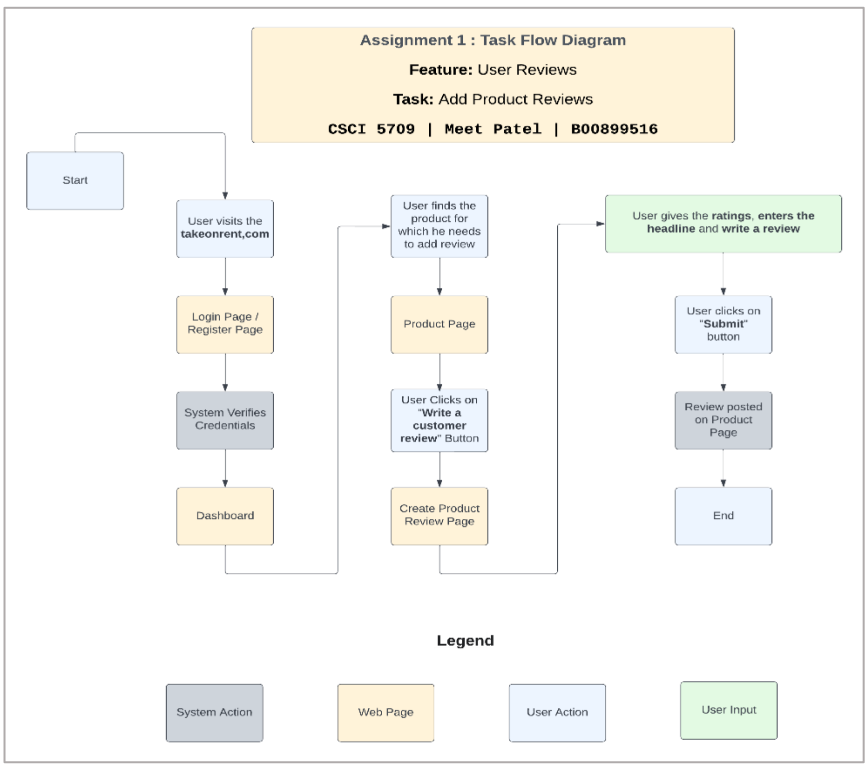


Figure 53: Add Product Review Task Flow

Task 2: Add Owner Review

Scenario: The user recently rented a product from one of the owners who rented his product on TakeOnRent website. Now the user wants to share his experience with others so that they get an idea about the owner's behaviour and the services provided by the owner.

Use Case:

Assume that the user is a registered user and has rented one of the products from one of the owners form the website.

1. Users open the browser and visit "TakeOnRent.com"
2. User signs in to his account
3. User then opens the product that he already rented
4. In the customer review section, the user clicks on a button which is named "leave owner feedback"
5. A new page opens where the user gives an overall rating and writes the owner review
6. User then submits the review
7. Finally, the review will be posted on the Owner page.

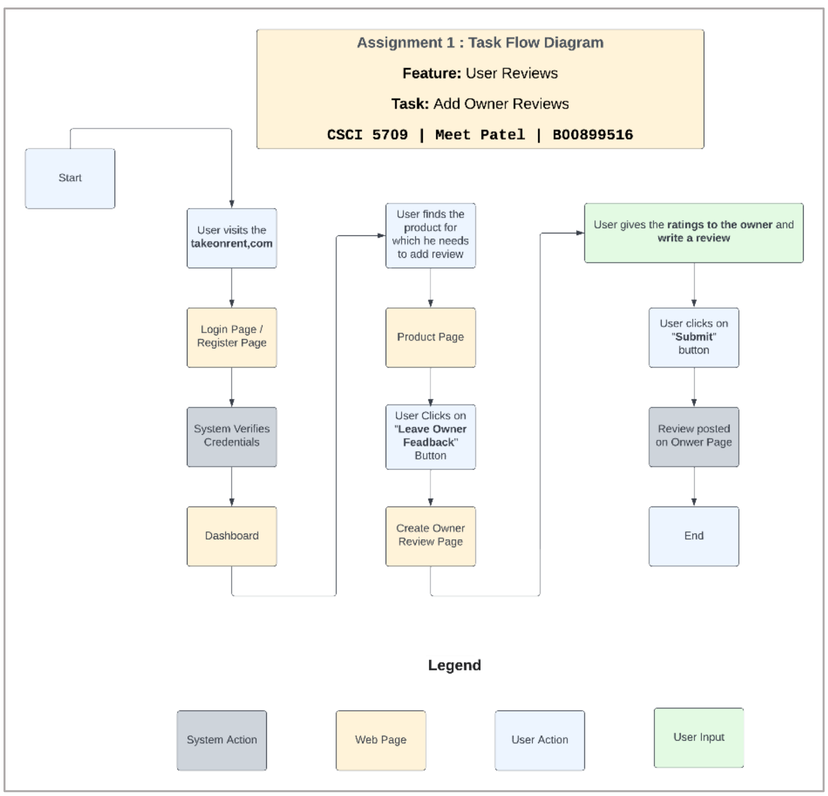


Figure 54: Add Owner Review Task Flow

Task 3: Update Posted Review

Scenario: The user recently posted an owner and product review on TakeOnRent Website. Now user wants to update one of the posted product/owner reviews.

Use Case:

Assume that the user is a registered user and has rented one of the products from one of the owners from the website and already rented one of the products from the website and a wrote product or owner review

1. Users open the browser and visit "TakeOnRent.com"
2. User signs in to his account
3. User then opens the posted reviews page
4. User clicks on the update button to update the posted review
5. A page will open where the user will be able to update the review
6. User then clicks on submit
7. Finally, the review will be posted on the product page

Task 4: Delete Posted Review

Scenario: The user recently posted an owner and product review on TakeOnRent Website. Now user wants to delete one of the posted product/owner reviews.

Use Case:

Assume that the user is a registered user and has rented one of the products from one of the owners from the website and already rented one of the products from the website and wrote product or owner review

1. Users open the browser and visits "TakeOnRent.com"
2. User sign in into his account
3. User then opens the posted reviews page
4. User clicks on delete button to delete the posted review
5. Finally, the review will be deleted on the product/owner page

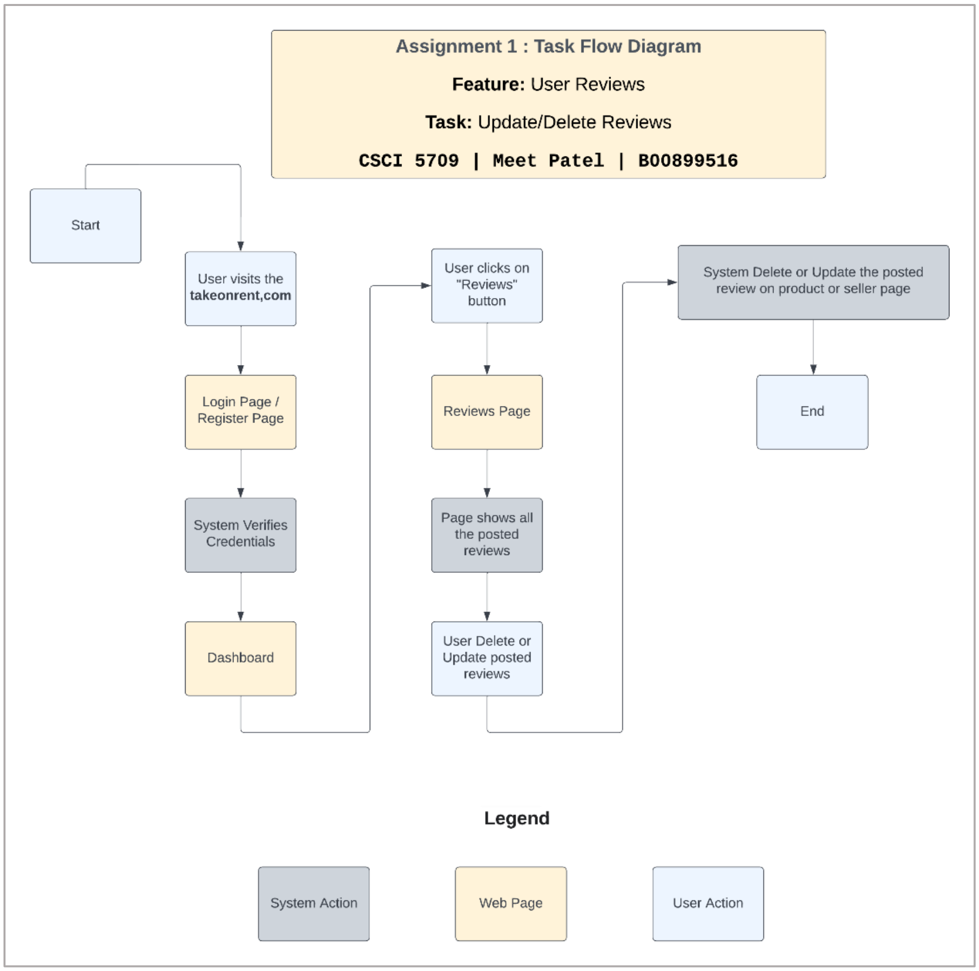


Figure 55: Update/Delete Task Flow

4. Analytics

Task 1: Perform Analysis

Scenario: Suppose the owner listed 100 products and wants to know how many products are currently on rent and how many products are not rented. Then the analytics feature comes into the picture. Using analytics owners will be able to know how many products are on rent vs products which are not rented. In addition, analytics is also responsible for displaying the graphs such as months vs products rented, months vs users, product categories vs products rented and many such charts.

Use Case:

Assume that the user/owner is registered.

1. Users open the browser and visit “takeonrent.com”
2. User signs in to his account

3. The user then opens the Analytics page from a link given in the header

The user then performs the analysis as required

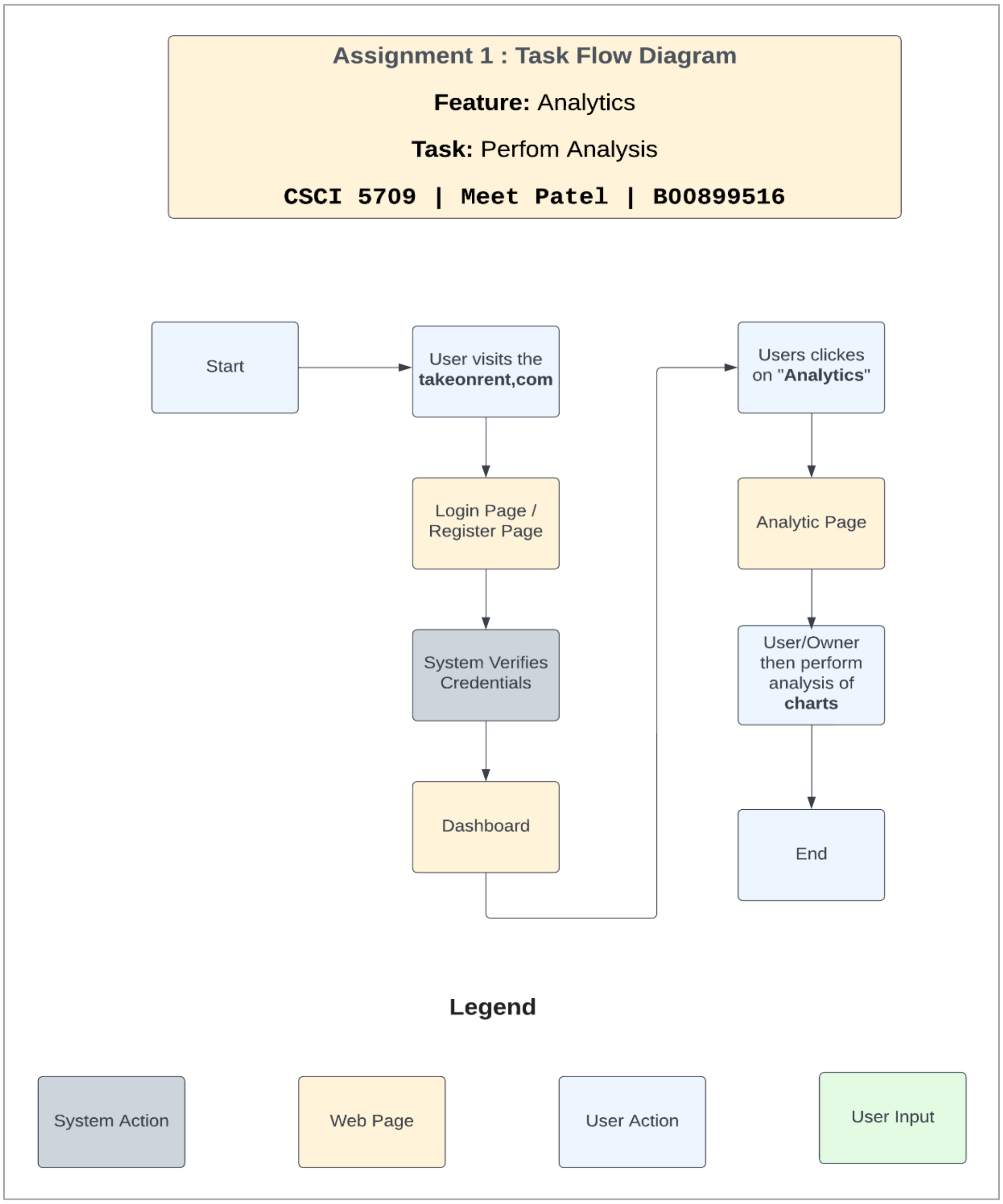


Figure 56: Analytics Task Flow

5. Community Forum

It is expected that the user has signed up on TakeOnRent website and all the use cases for User Management are satisfied.

<u>Task 1: Create a post on the Community forum.</u>

Scenario:

The owner wants to notify publicly about the equipment he wants to rent. He navigates to the community forum page and creates a post with the equipment description and renting period to get a better reach.

Use Case:

1. User opens browser and visits the TakeOnRent website.
2. System displays login page.
3. User successfully logs in and navigates to the community forum page.
4. User clicks on the "CREATE" button on the community forum page.
    1.1 System displays a form with the required fields.
    1.2 User enters the information requested by the system and hits submit button.
    1.3 System highlights the fields which was left blank and presents a message to re-enter the data for those fields.
    1.4 User re-enters the information and the system will re-validate it.
    1.5 In case of invalid information, system will again ask the user to re-enter the invalid data until the user enters valid data or the user clicks on Cancel button.
2. If the data entered by user is valid, the system will create a post on the community forum page for everyone to view it.

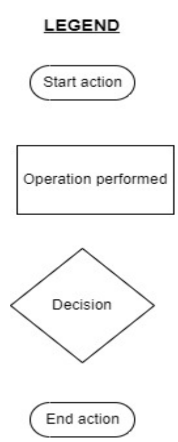


Figure 57: Legend

Task Flow diagram:

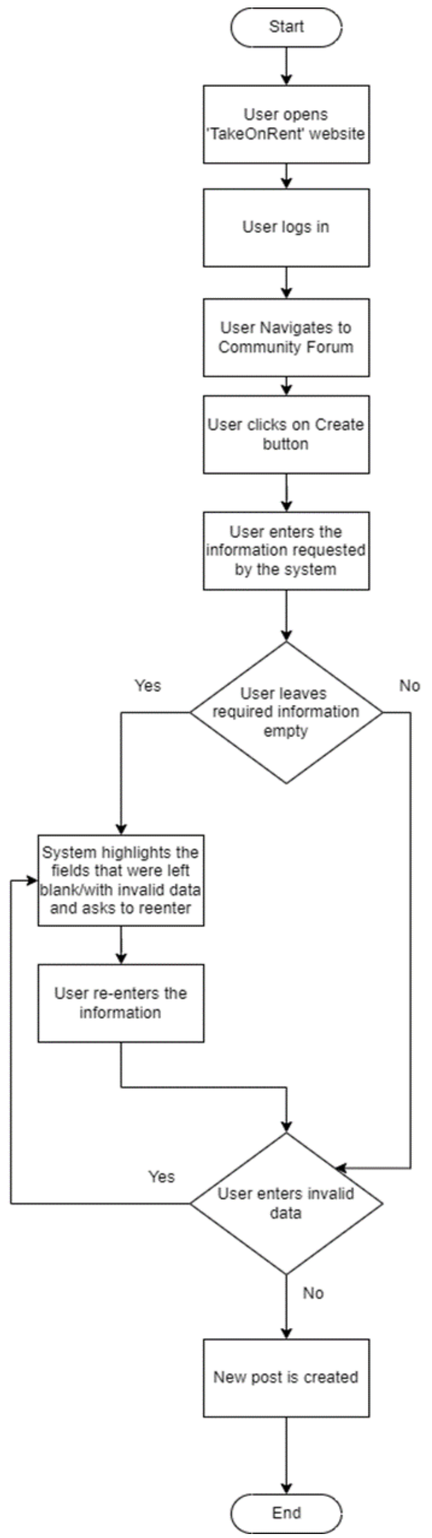


Figure 58: Create Post Task Flow Diagram

Task 2: Update post on Community forum

Use Case:

1. User opens browser and visits the TakeOnRent website.
2. System displays login page.
3. User successfully logs in and navigates to the community forum page.
4. System displays all the posts created by any user on the page.
    4.1 The logged in user has not created any post so he cannot view the vertical options button on any post on displayed on the forum.
    4.2 The logged in user creates a post and it is expected that all the use cases for the Create Post on Community forum are satisfied.
    4.3 The user reloads the community forum page and is now able to see his post on the community forum.
5. The user clicks on the vertical options button on the top right corner of the post created by him.
6. The system displays Edit and Delete options to the user.
7. The user clicks on Edit option.
8. The system displays the post as editable where the user can change all the information of the post which were asked at the time of creation of the post.
    8.1 User enters the information he wants to update, and hits submit button.
    8.2 System highlights the data which was left blank and presents a message to re-enter the data for those fields.
    8.3 User re-enters the information and the system will re-validate it.
    8.4 In case of invalid information, system will again ask the user to re-enter the invalid data until the user enters valid data or the user clicks on Cancel button.

9 If the data entered by user is valid, the system will update the post on the community forum page for everyone to view it.

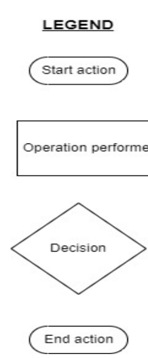


Figure 59: Legend

Task Flow Diagram:

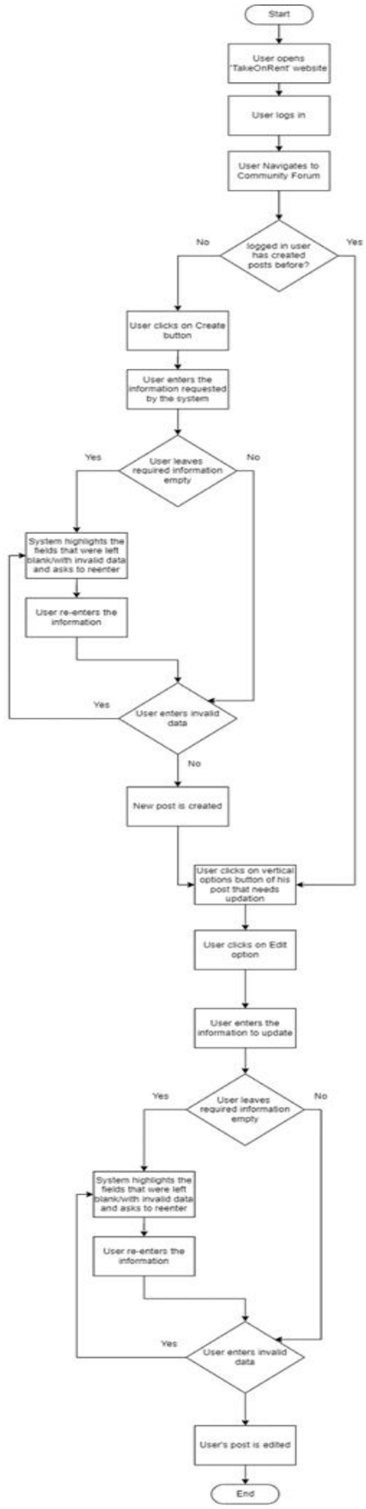


Figure 60: Update Post Task Flow Diagram

Task 3: Delete post on Community forum

Use Case:

1. User opens browser and visits the TakeOnRent website.
2. System displays login page.
3. User successfully logs in and navigates to the community forum page.
4. System displays all the posts created by any user on the page.
   - 4.1 The logged in user has not created any post so he cannot view the vertical options button on any post on displayed on the forum.
   - 4.2 The logged in user creates a post and it is expected that all the use cases for the Create Post on Community forum are satisfied.
   - 4.3 The user reloads the community forum page and is now able to see his post on the community forum.
5. The user clicks on the vertical options button on the top right corner of the post created by him.
6. The system displays Edit and Delete options to the user.
7. The user clicks on Delete option.
8. The system deletes the post from the community forum page, and no one can now view this post.

Task Flow Diagram:

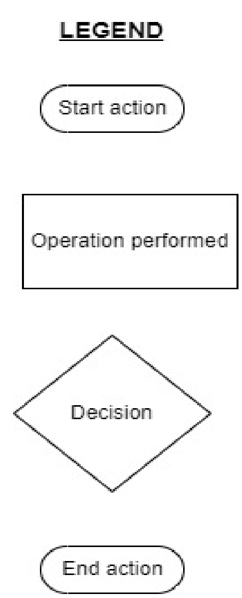


Figure 61: legend

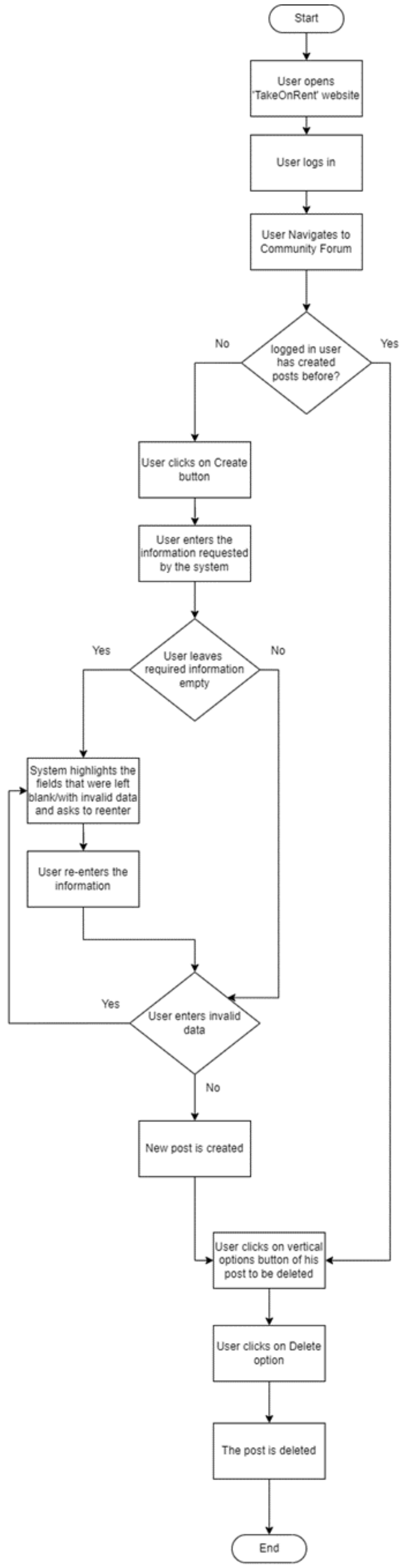


Figure 62: Delete Post Task Flow Diagram

Task 4: Filter all posts on Community forum.

Use Case:

1. User opens browser and visits the TakeOnRent website.
2. System displays login page.
3. User successfully logs in and navigates to the community forum page.
4. System displays all the posts created by any user on the page with a filter option on the top right corner of the website.
   4.1 User clicks on Filter select button.
   4.2 System gives three options to filter by "My Posts", "Date" or by "Category".
      4.2.1 User clicks on My Posts.
         4.2.1.1 The logged in user has not created any post so he cannot view the vertical options button on any post on displayed on the forum.
         4.2.1.2 The logged in user creates a post and it is expected that all the use cases for the Create Post on Community forum are satisfied.
         4.2.1.3 The user reloads the community forum page and is now able to see his post on the community forum.
         4.2.1.4 The user clicks on Filter and selects the option "My Posts".
         4.2.1.5 System filters the posts in such a way that the logged in users' posts are listed first.
      4.2.2 User clicks on Date.
      4.2.3 System filters the posts in such a way that the latest posts are listed first.
      4.2.4 User clicks on Category.
      4.2.5 System filters the posts in such a way that the all the posts can be viewed with all the categories grouped together.

Task Flow diagram:

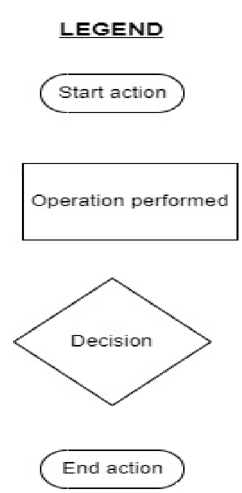


Figure 63: legend

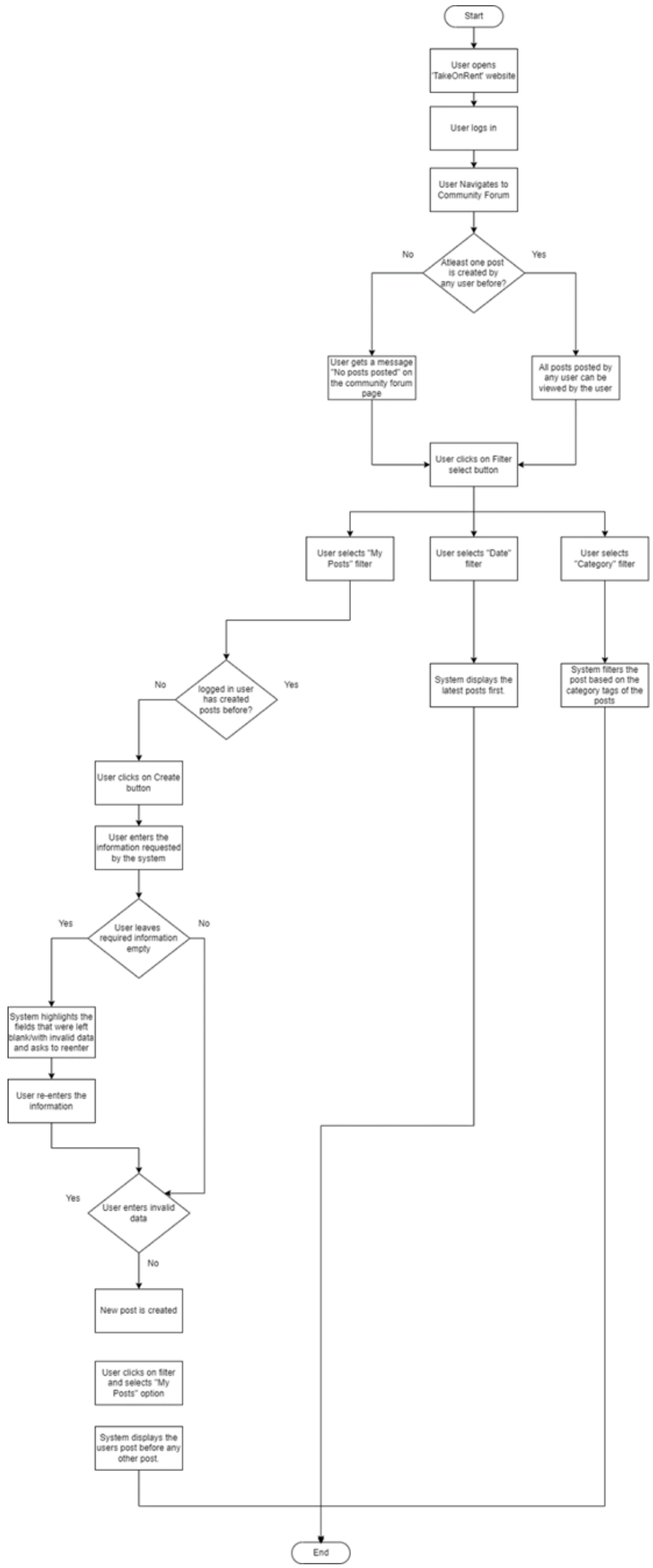


Figure 64: Filter Posts Task Flow Diagram

Task 5: View all posts on Community forum

Use Case:

1. User opens browser and visits the TakeOnRent website.
2. System displays login page.
3. User successfully logs in and navigates to the community forum page.
4. System displays all the posts created by any user on the page and the user can view the posts.

Task Flow diagram:

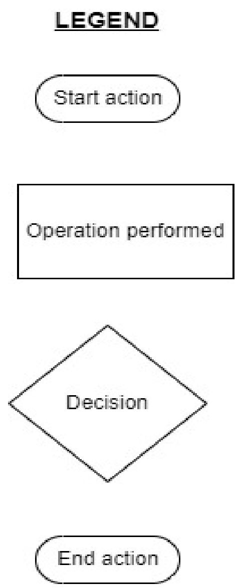


Figure 65: legend

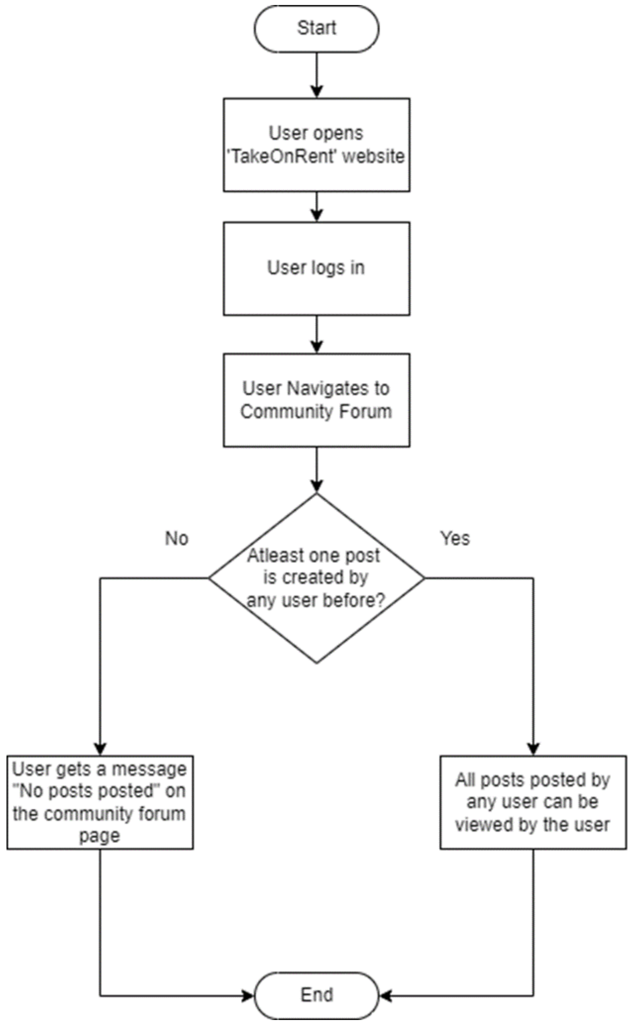


Figure 66: View Posts Task Flow Diagram

6. Messaging Service

It is expected that the user has signed up on TakeOnRent website and all the use cases for User Management are satisfied.

Task 1: *Send a message to the owner*

Scenario:
A user while navigating the products wants to contact the owner of a particular product. The user clicks on the messaging button for a particular product for sending a message to the registered email of the owner.
Use Case:

1. User opens browser and visits the TakeOnRent website.
2. System displays login page.
3. User successfully logs in and navigates to the Products page.
4. System displays list of products available to rent.
5. User clicks the messaging button on the bottom right part of the product.
    1. System displays a form to send the message to owner with required fields.
    2. User enters the information requested by the system and hits submit button.
    3. System highlights the fields which was left blank and presents a message to re-enter the data for those fields.
    4. User re-enters the information and the system will re-validate it.
    5. In case of invalid information, system will again ask the user to re-enter the invalid data until the user enters valid data or the user clicks on Cancel button
6. If the data entered by user is valid, the system will send a message to the registered email of the owner.

Task Flow diagram

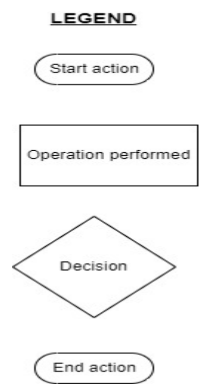


Figure 67:legend

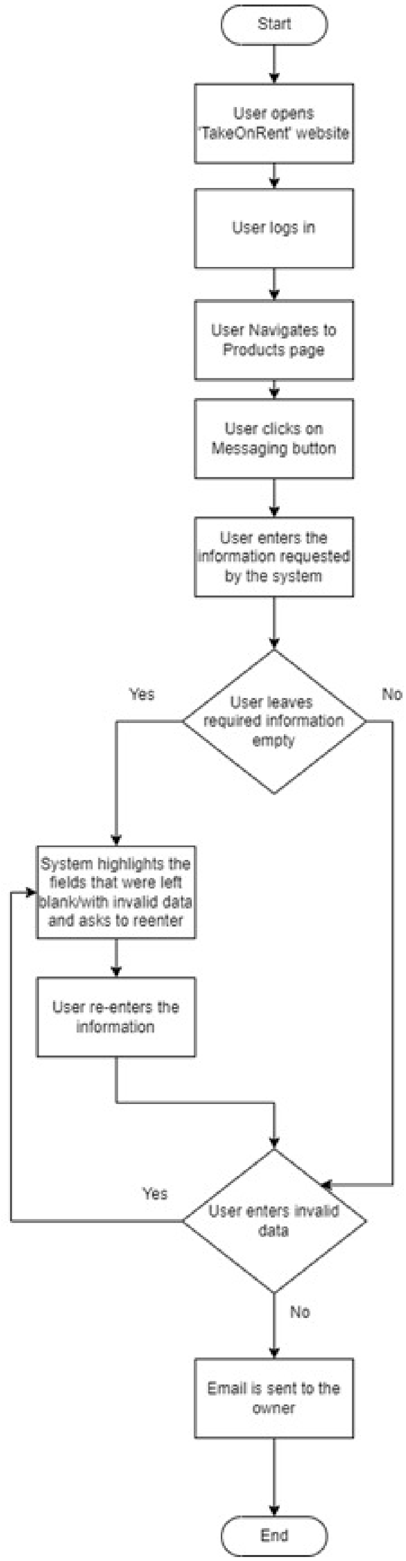


Figure 68: Messaging Task Flow

7. Payment Gateway

Task 1: Add Information

Scenario: This feature will be used, when the new user creates an account to put any useful equipment on rent.

- **User Persona:** The user who wants to rent equipment and wants to pay for it.
- **Need:** To allow the user to rent equipment and to process the payment easily using their online payment methods such as debit or credit cards
- **Context:** User wants to rent equipment and doesn't to visit the store in order to pay instead they want to send someone else just to pick it up and hence they want to pay it online with their convenience from the online payment portal

Use case:

1. Users click on the Payment Gateway Page
2. User then click on the add new address radio button
3. User then fills up the information mentioned such as First Name, Last Name, Address, and Email address
4. Then they will be asked for the Phone number
5. Then they click on Add Address button below that.

Task Flow

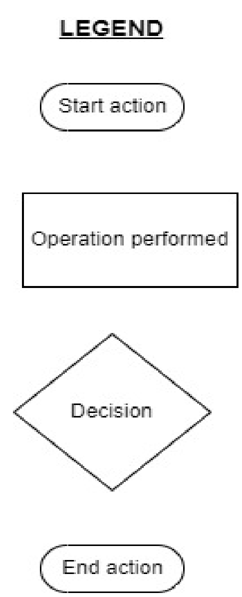


Figure 69: legend

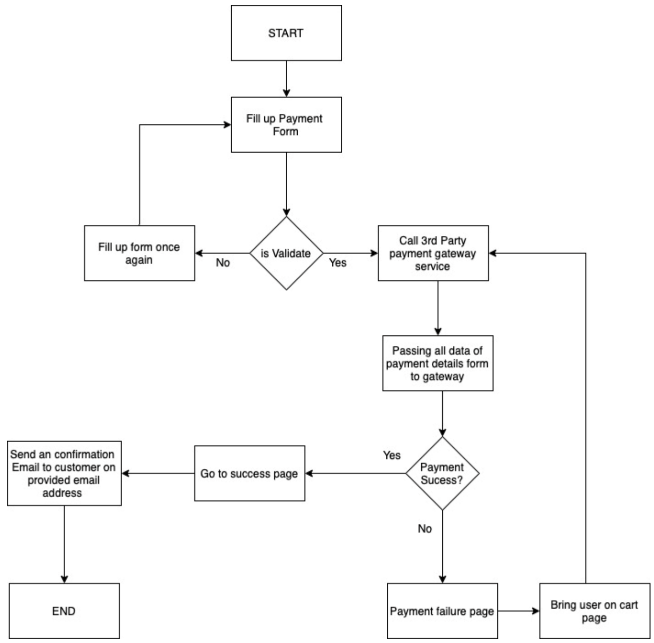


Figure 70: User information fill up Flow Diagram

Task 2: Payment Processing

Scenario: This feature will be used, when the user wants to select the payment method and pay

- **User Persona:** The user who thinks of paying for the rental equipment at their convenience they use this portal
- **Need:** For the seamless payment from the user.
- **Context:** A user who wants to use their credit, debit, PayPal, etc. portal to make a payment for their rent

Use case:

1. Users go to the Payment Gateway Page
2. We will ask the method of Payment they want to use like Paypal, Mastercard, Visa, etc.
3. Then we will ask for their card details

Task Flow:

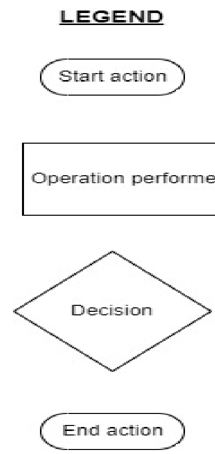


Figure 71: legend

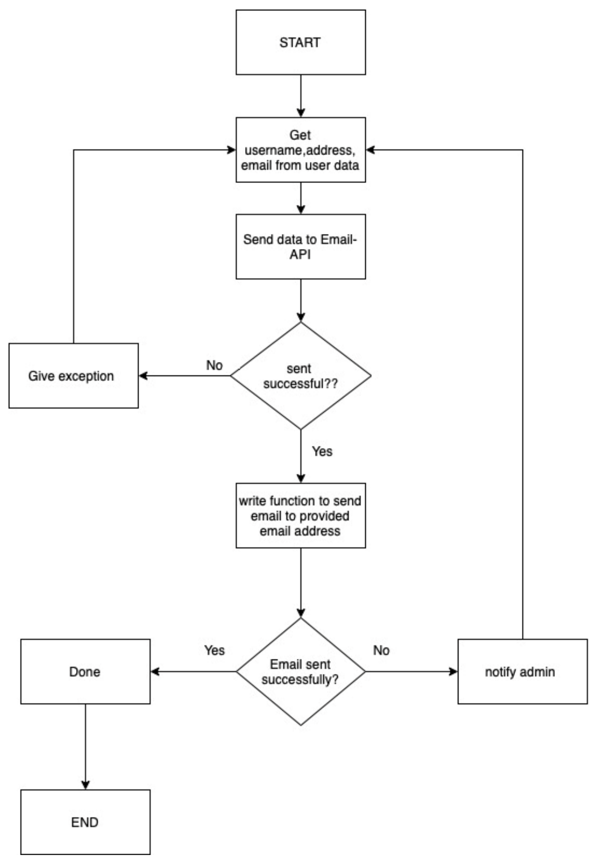


Figure 72:User Payment Processing Flow Diagram

Task 3: Confirmation

Scenario: This feature will be used, when users want to get the details about their payment and services

- **User Persona:** The user wants to take equipment on rent but also want the confirmation of their service and charges
- **Need:** To have a record of their service and payment
- **Context:** The user visits the website purchase or rent the equipment but then he wants to know what all charges he has been charged for.

Use case:

1. User goes to the payment Gateway page

   https://chipper-manatee-358ab9.netlify.app/

2. Then they click on proceed to book
3. The page will show all the charges and then user click on confirm
4. It will send a receipt with all the details to the user email address.

Task Flow:

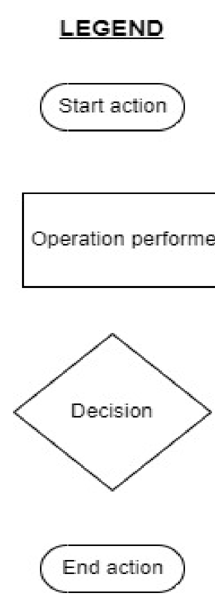


Figure 73: legend

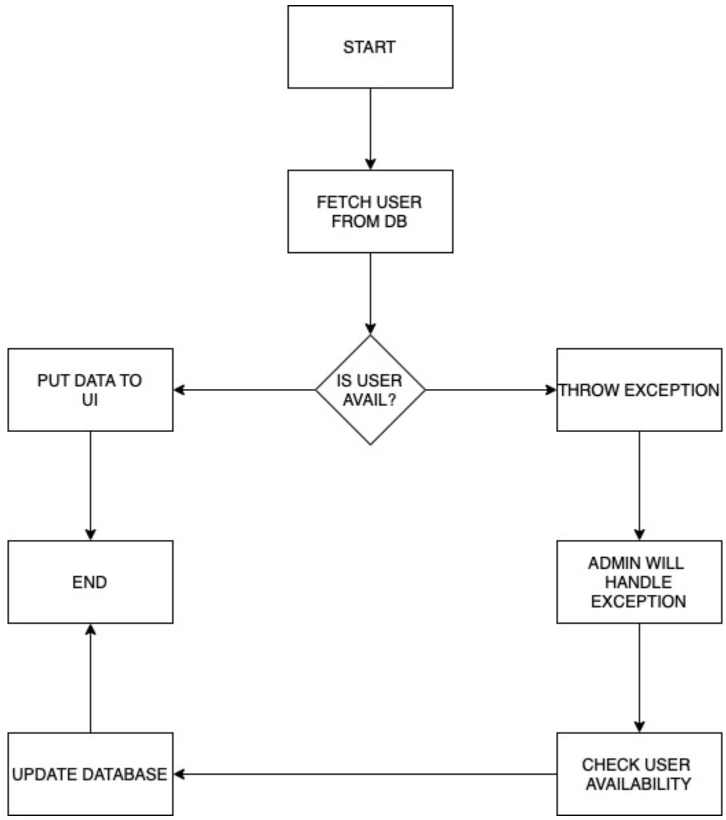


Figure 74:Payment confirmation Flow Diagram

8. Insurance

In this feature, it will fetch all the products selected, and it will ask the user if they want the insurance or not; if they select the checkbox of insurance needed, then it will add 5% of each product price, and it will show up in the billing as an insurance price and add in the cost summary.

Task 1: Add Information

Scenario: This feature will be used, when the new user creates an account to put any useful equipment on rent.

- **User Persona:** The user who wants to rent equipment and wants to pay for it.
- **Need:** To allow the user to rent equipment and to process the payment easily using their online payment methods such as debit or credit cards
- **Context:** User wants to rent equipment and doesn't to visit the store in order to pay instead they want to send someone else just to pick it up and hence they want to pay it online with their convenience from the online payment portal

Use case:

6. User click on the Payment Gateway Page

   https://chipper-manatee-358ab9.netlify.app/

7. User then click on the add new address radio button
8. User then fills up the information mentioned such as First Name, Last Name, Address, and Email address
9. Then they will be asked for the Phone number
10. Then they click on Add Address button below that.

Task Flow

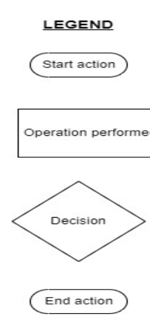


Figure 75: legend

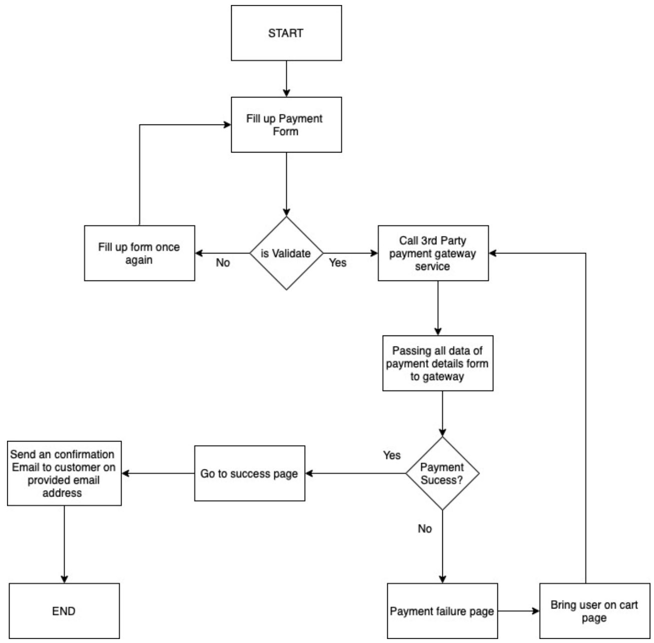


Figure 76: User information fill up Flow Diagram

Task 2: Payment Processing

Scenario: This feature will be used, when the user wants to select the payment method and pay

- **User Persona:** The user who thinks of paying for the rental equipment at their convenience they use this portal
- **Need:** For the seamless payment from the user.
- **Context:** A user who wants to use their credit, debit, PayPal, etc. portal to make a payment for their rent

Use case:

1. Users go to the Payment Gateway Page
2. We will ask the method of Payment they want to use like Paypal, Mastercard, Visa, etc.
3. Then we will ask for their card details

Task Flow:

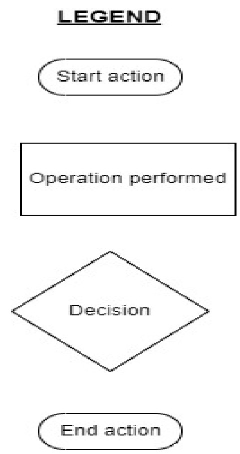


Figure 77: legend

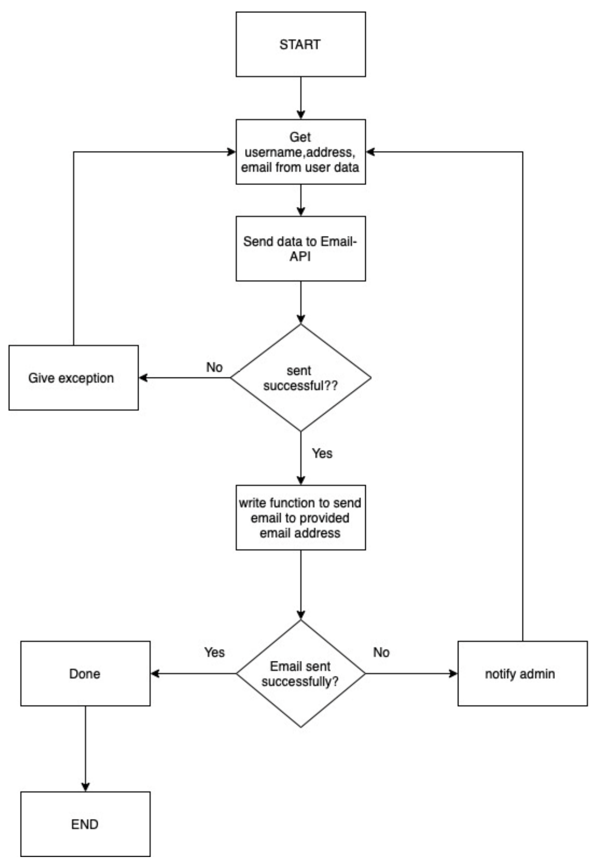


Figure 78:User Payment Processing Flow Diagram

Task 3: Confirmation

Scenario: This feature will be used, when users want to get the details about their payment and services

- **User Persona:** The user wants to take equipment on rent but also want the confirmation of their service and charges
- **Need:** To have a record of their service and payment
- **Context:** The user visits the website purchase or rent the equipment but then he wants to know what all charges he has been charged for.

Use case:

1. User goes to the payment Gateway page
2. Then they click on proceed to book
3. The page will show all the charges and then user click on confirm
4. It will send a receipt with all the details to the user email address.

Task Flow:

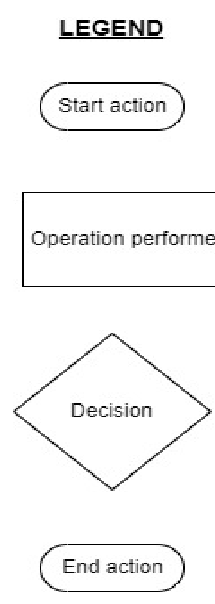


Figure 79: legend

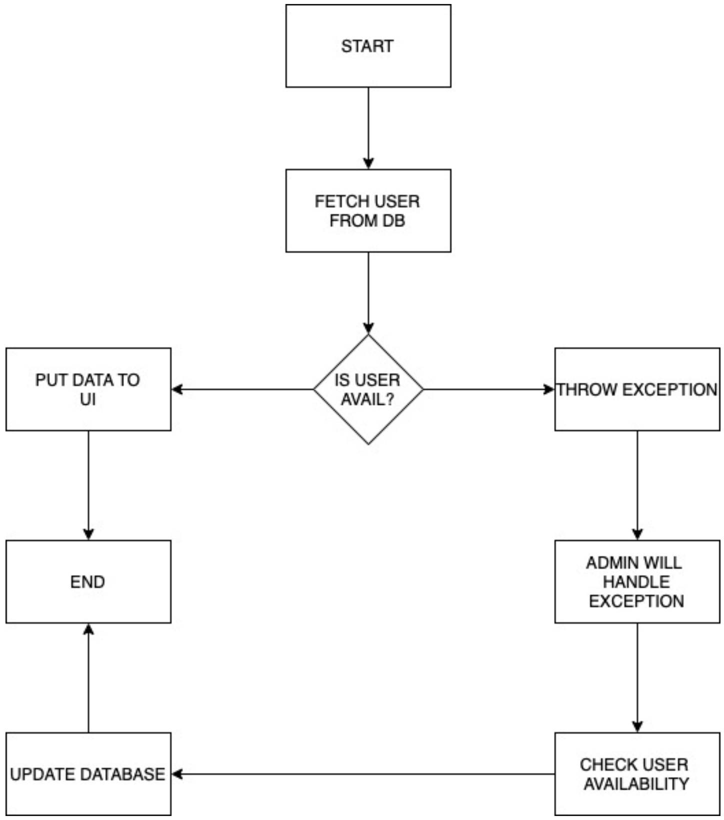


Figure 80:Payment confirmation Flow Diagram

9. Put on rent

Task 1: Add Equipment

Scenario: An owner (seller) wants to add the equipment on an online platform so that a renter can contact and rent it.

- User persona: Owner
- Need: Upload the product to give it on rent to the customer
- Context: The owner uploads the product so that customers can rent the required equipment by contacting the owner using the web app.

Use case:

1. The owner needs to create an account and register if using the application for the first time and enter the required information.
2. Once done, a confirmation mail is sent to the user for account creation.
3. If the user has already one, then the user needs to log in with the correct information.

4. The first page displayed will be the product catalog, where the user can see the home page along with the message “Welcome, user”.
5. Then the owner needs to select the option for “Add equipment”.
6. Once selected, it redirects to a page wherein the information for the equipment is to be added in the form.
7. Equipment name, category, price, image, description, location, quantity, security deposit amount, rent amount, and available for are the fields the which owner will fill for the equipment.
    a. If the owner does miss out on any of the mandatory fields, then it will show to fill that field.
8. After filling in the data in a correct way, when owner clicks submit, a message for “Equipment uploaded successfully” is displayed.

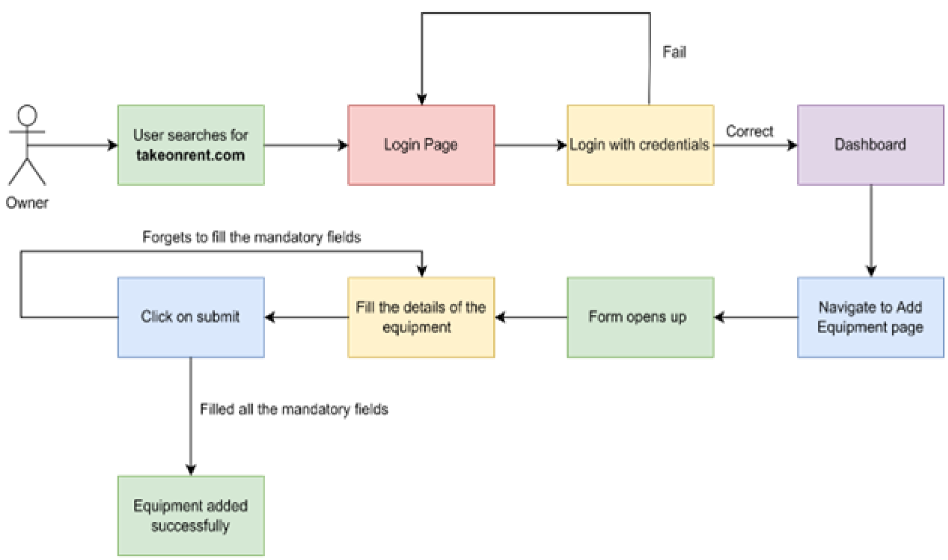


Figure 81: Add Equipment task Flow

Task 2: View and Edit Equipment

Scenario: An owner can view and edit the equipment in case of change in any of the parameters of the equipment including price, status, quantity, description, etc.

- User Persona: Owner
- Need: Equipment can be given on rent as per requirement
- Context: Viewing and updating the product keeps the customer updated so that owner can give it on rent in an easy way.

Use case:

1. The owner needs to create an account and register if using the application for the first time and enter the required information.
2. Once done, a confirmation mail is sent to the user for account creation.
3. If already have one, then the user needs to log in with the correct information.
4. The first page displayed will be the product catalog, where the user can see the home page along with the message “Welcome, user”.
5. Then the owner needs to search for the specific equipment and can view the equipment.
6. The next step is to click the “Edit equipment” button.
7. Once clicked, it redirects to a page wherein the information for the equipment is already field and each field is editable.
8. Equipment name, category, price, image, description, location, quantity, security deposit amount, rent amount, and available for are the fields that which owner will be able to update anytime.
9. After updating the required data, when the owner clicks submit, a message for “Equipment updated successfully” is displayed.

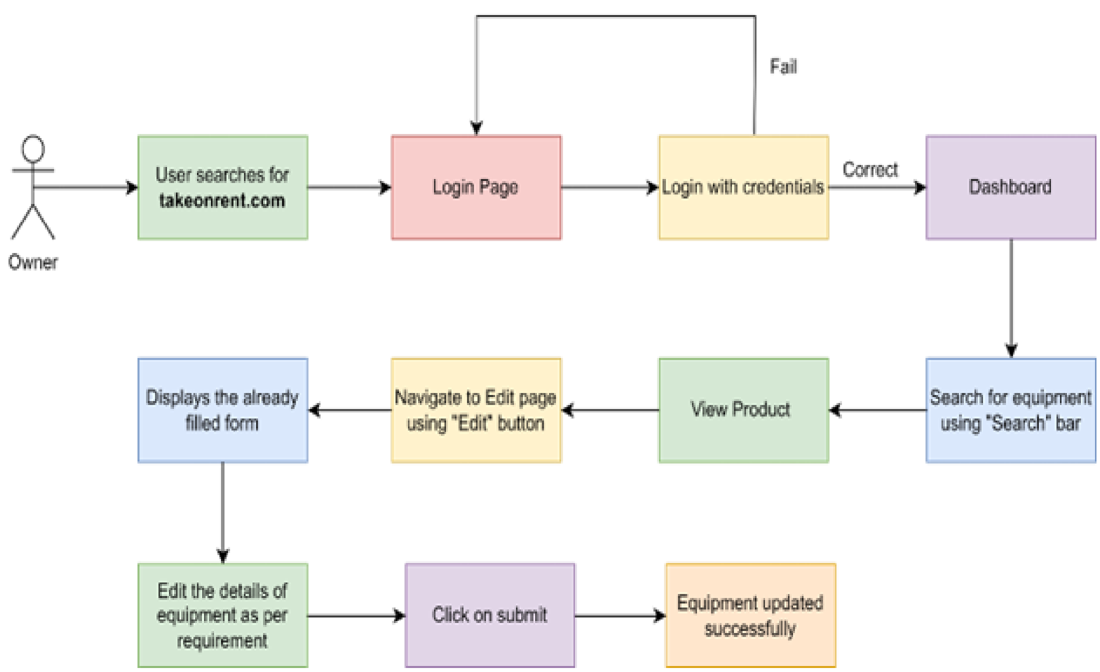


Figure 82: View and Edit equipment info task Flow

<u>Task 3: View and Delete Equipment</u>

Scenario: An owner wants to delete the equipment as it is no longer to use from the online rental platform.

- User Persona: Owner
- Need: If the equipment cannot be given on rent anymore, it needs to be removed.
- Context: If the owner feels that equipment is broken or damaged, then it can be removed wholly using the web app to avoid it being given on rent.

Use-case:

1. The owner needs to create an account and register if using the application for the first time and enter the required information
2. Once done, a confirmation is sent to the user for account creation.
3. If already have one, then the user needs to log in with the correct information.
4. The first page displayed will be the product catalog, where the user can see the home page along with the message “Welcome, user”.
5. Then the owner needs to search for the specific equipment and can view the equipment.
6. Then the owner needs to select the option for “Delete equipment”.
7. Once selected, it redirects to a page wherein the information for the equipment is to be deleted.
8. Once the user reaches the page, whether the user needs to click the button “Delete”.
    a. A pop-up is raised whether user wants to finally delete the equipment or not.
    b. If the user clicks yes then a message is displayed, “Equipment deleted successfully”.
    c. If the user clicks no, then he remains on the same page.
9. The equipment gets successfully deleted from the main page.

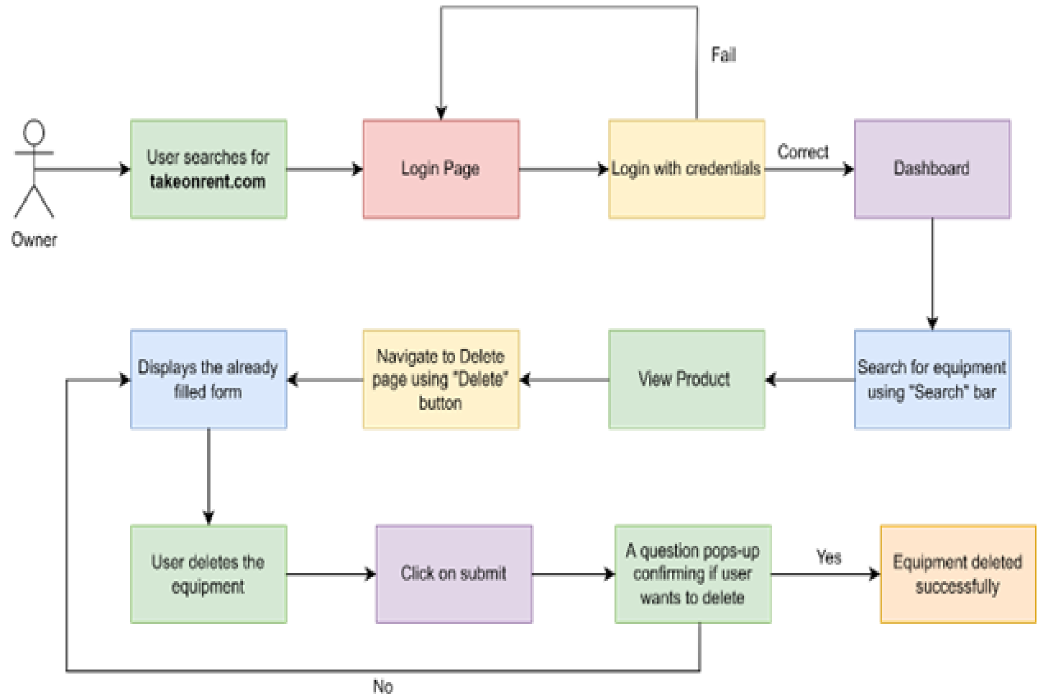


Figure 83: View and Delete Equipment ***(Rutvik Patel)***

10. Coupon Management

Task 1: Apply Coupon

Scenario: An owner wants to attach a coupon with an equipment's amount to give a discount to the customer renting that equipment.

- User persona: Owner
- Need: Giving a discount to the customer renting an equipment
- Context: The owner can decide which equipment can be put on discount to give it rent at the best price to the customer.

Use case:

1. The owner needs to create an account and register if using the application for the first time and enter the required information.
2. Once done, a confirmation is sent to the user for account creation.
3. If already have one, then the user needs to log in with the correct information.
4. The first page displayed will be the product catalog, where the user can see the home page along with the message "Welcome, user".
5. Once the owner adds product information, then upon searching an option for "Add coupon" is given.

6. Once selected, it redirects to a page wherein the information for the equipment is displayed and coupons can be added/removed.
7. The user needs to enter the equipment id and from that user will be able to add the discount amount.
8. After attaching the coupon, a pop-up is displayed, wherein confirmation for the application of coupon is asked.
    a. If the user selects “yes” when he clicks on the submit button, a question pops up asking for confirmation for the submission.
    b. If the user selects “no” then he remains on the same page

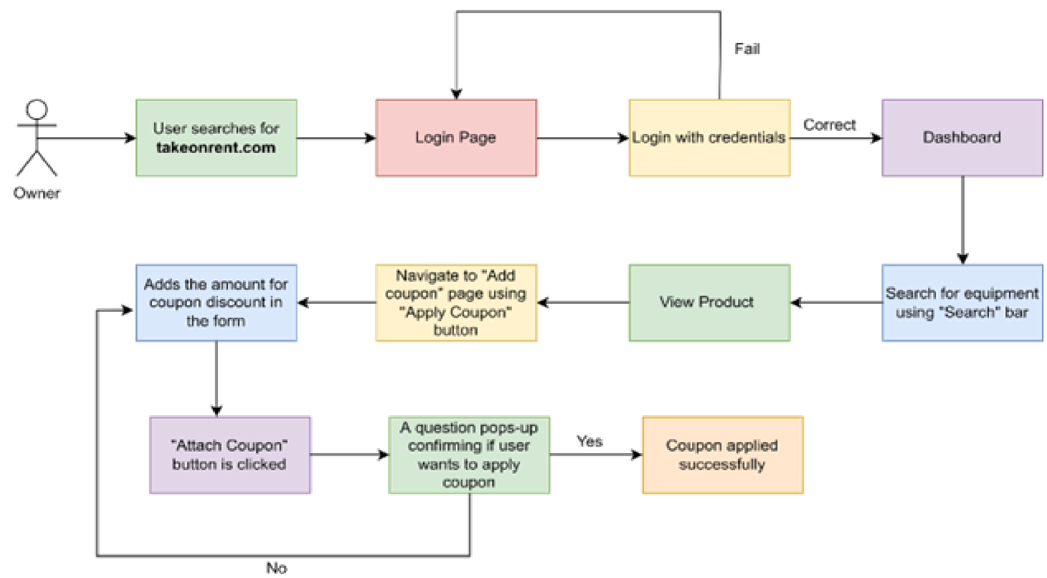


Figure 84: Apply coupon on Equipment Task Flow (Rutvik Patel)

11. Cart Management

Task 1: Add product to the cart Scenario:

The renter is interested in the product displayed in the product catalog. To save the product for future reference the renter decides to add the product to the cart.

Persona: Renter

Feature: Cart Management

Need: Add the product to the cart

Context: The renter likes a product detail, he/she wants to add in a cart for future reference feature to save it

Use Case:

1. Add the product to the cart or increase the product quantity.
2. The renter logs into our application.
3. System displays the home page of our application.
4. The renter selects the product from the product catalog.
5. The system verifies if the selected product is already added to the cart or not. The system displays the “Add to cart” option if the post is not added to the cart. The system displays the “Remove from cart” option if the product is already added to the cart.
6. The renter selects the “Add to cart” option if the selected post is not added to the cart section.
7. System displays the “Added to cart” message after the product is added to the cart by the system.
8. The renter clicks on the “cart” option provided to view all the products added to the cart.
9. The system displays the products saved from the cart.
10. The renter sees the cart where all the saved products are listed.
11. The user increases the product quantity from the cart using the “+” button if wanted.
12. The user checkout and navigate to the payment page

Task Flow:

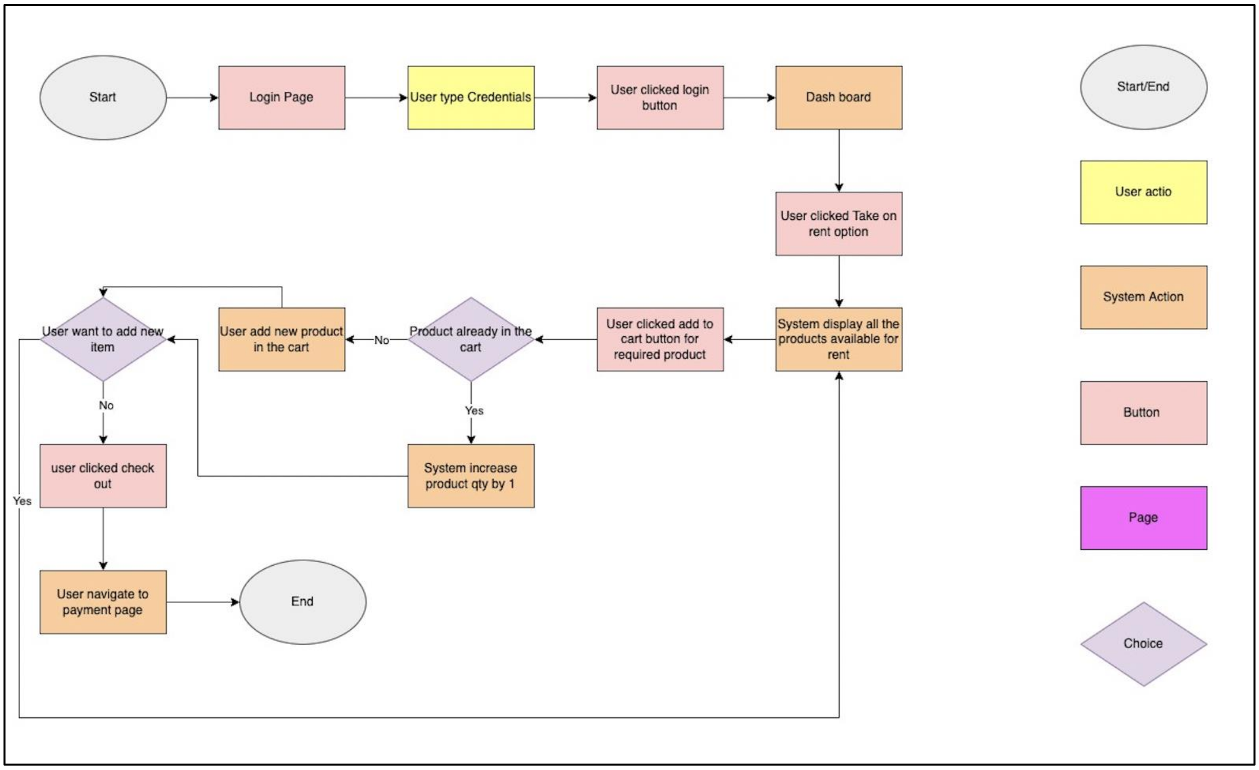


Figure 85: Add product to cart task flow

Task 2: Remove from the Cart or drop the product quantity.

Scenario: The renter is no longer interested in the product that he/she added to the cart, so he/she removes it from the cart.

Persona: Renter

Feature: Cart Management

Need: Remove the products from the cart

Context: The renter no longer likes the product, he/she uses the remove from cart to remove it

Use Case: Remove the product from the cart

1. The renter logs into our application.

2. System displays the home page of our application.

3. The renter selects the product from the Product catalog.

4. The system takes the user to the cart.

5. User decreases the quantity of the product by 1.

6. The system verifies If it is the last item in the Cart or not. If not, then the system decreases the item qty by 1 and displays “One item removed from the cart.

7. If it is the last item then the System removes the last item and displays “product removed from the cart”.

8. If the user wants to continue removing the item, then, go to step 4 again.

9. otherwise user checkout and navigate to the payment page.

Task flow:

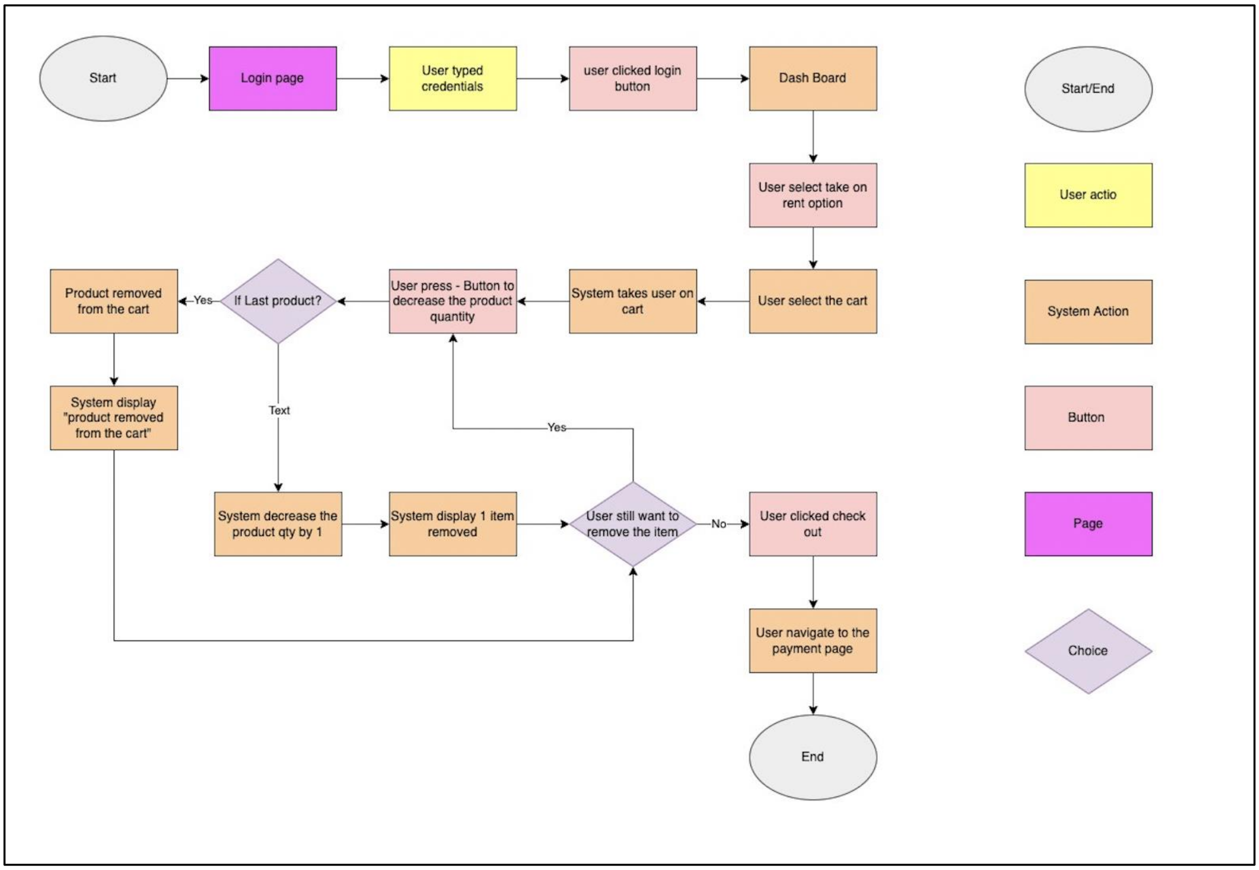


Figure 86: Drop product from cart task flow

12. Product Catalogue

Task 1: Filter product by Name:

Scenario:

A user while navigating the product wants to filter the product using the product name. A user clicks on the Product image to see the product details and to add a product to the cart.

Use Case:

1. User Open the browser and go to the TakeOnRent website.
2. The system displays the login page.
3. The user successfully logs in.
4. The user navigates to the products page.
5. System display s list of products available for rent.
6. The user enters the product name on the search bar.
   6.1 system display all the products which match the name entered by the user.
   6.2 The user clicks on the required product.
   6.3 The system displays the product name, price, and description on a single product page.
   6.4 Users click on add product button to add the product to the cart.
7. System Add the product selected by the user to the cart.

Task Flow Diagram:

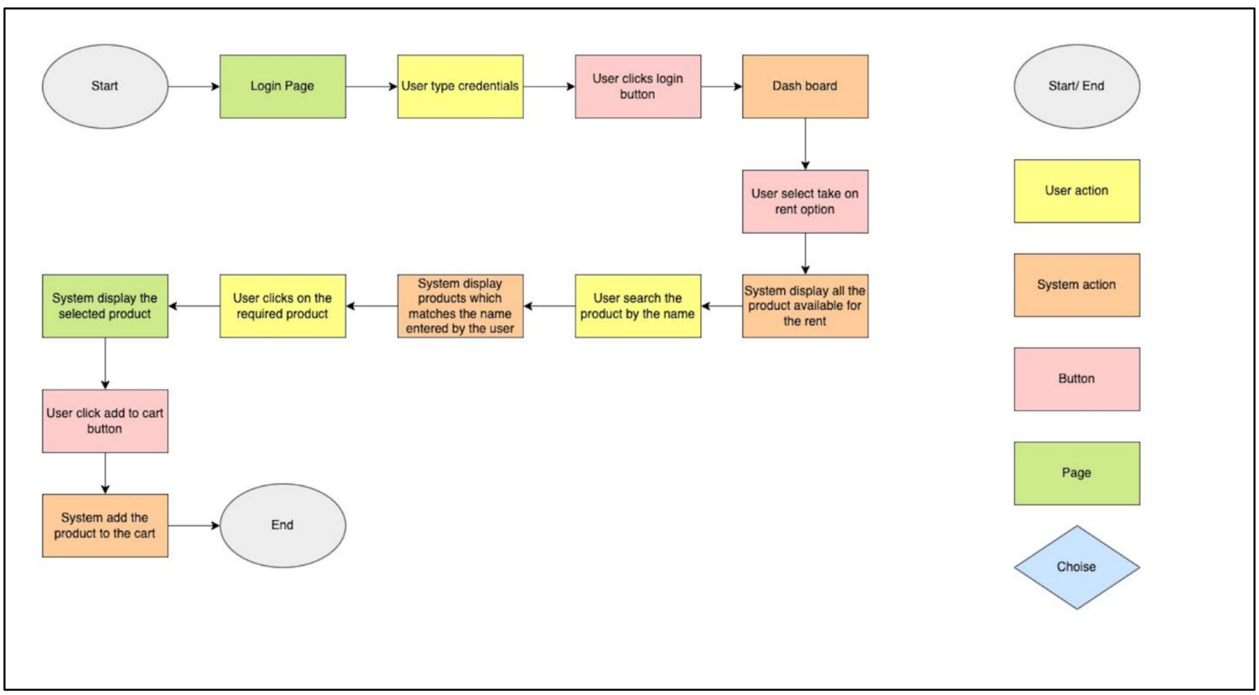


Figure 87: Filter product by Name Task Flow

Task 2: Filter product by Price:

Scenario:

A user while navigating the product wants to filter the product using product price. A user clicks on the Product image to see the product details and to add a product to the cart.

Use Case:

1. User Open the browser and go to the TakeOnRent website.
2. The system displays the login page.
3. The user successfully logs in.
4. The user navigates to the products page.
5. The system displays the list of products available for rent.
6. The user enters the minimum product price on the search bar.
   6.1 system display all the products which have less product price than the minimum product price entered by the user.
   6.2 The user clicks on the required product.
   6.3 The system displays the product name, price, and description on a single product page.
   6.4 Users click on add product button to add the product to the cart.
7. System Add the product selected by the user to the cart.

Task Flow Diagram:

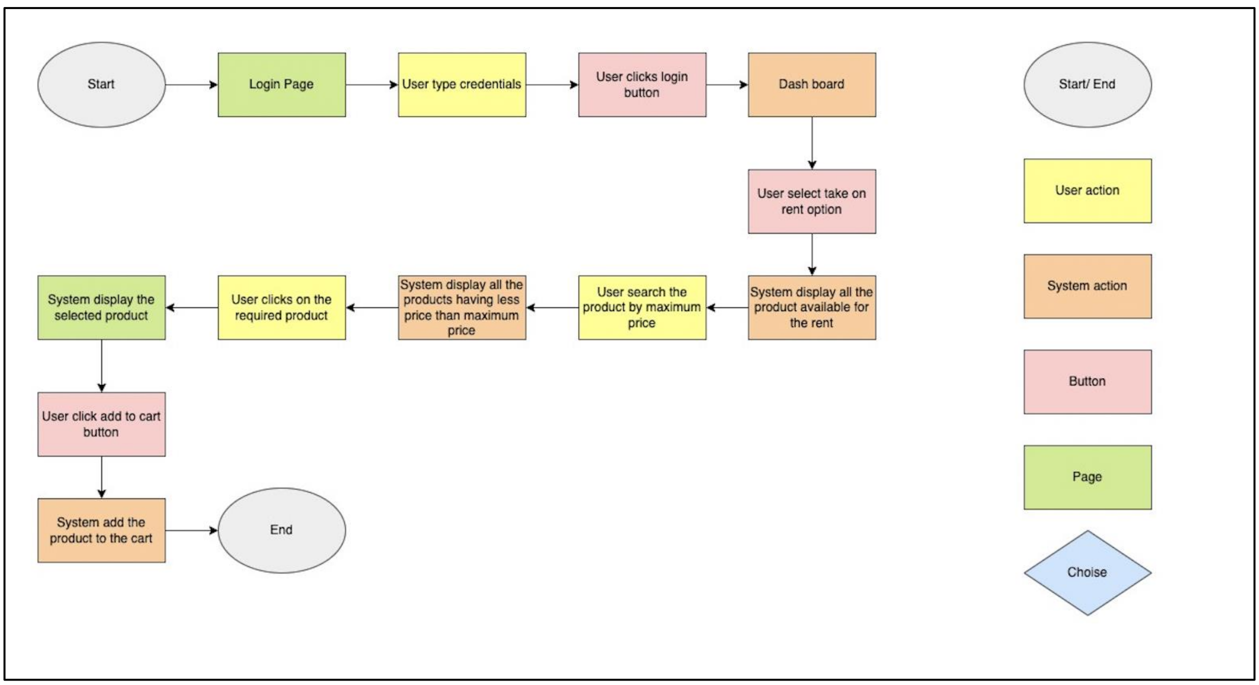


Figure 88: Filter product by Price Task flow

3. Asset inventory

Client-side languages:

The client-side languages, used in the development of the project "TakeOnRent" are:

- **React:** React is a JavaScript library for building user interfaces. It is used for performing validations for the front-end and making API requests to the server [1].
- **HTML:** HTML is the standard markup language used for developing web pages [2].
- **CSS:** CSS is the language used to apply styling to the HTML document. Here, ".css" files are used to give style to the elements [3].
- **Bootstrap:** Bootstrap is a free and open-source CSS framework used to make the website more responsive in front-end development. Here, inbuilt bootstrap classes can be used, and installed based on Reactjs, commonly referred to as "react-bootstrap" [4].
- **Material UI:** material UI is an open-source React component library, which includes a collection of "ready-to-use" prebuilt design components, which can be used to make the website's "look and feel" interactive and responsive [5].

Server-side languages:

For the server-side part of the project, the following technologies are used:

- **Node.js:** Node.js is an open-source, back-end JavaScript runtime environment used for server-side validations and access to the database for fetching the required data as per the received request from the client-side [6].
- **MongoDB:** MongoDB is a NoSQL database, used to store data in the form of documents. Data can be accessed using queries and perform CRUD operations [7].

Application Programming Interface (API):

Below mentioned is the API, which is used between server and client services:

- **REST API:** A RESTful API is used to communicate between the server-side and client-side by returning the current state of the requested resource by the server side, from the client side [8].

Images:

The format used for images to be uploaded is as below:

- JPG
- JPEG
- PNG

4. Group Roles

Name	Primary Role	Secondary Role
Vivek Patel	Full Stack Developer	Designer and Integrator
Rushi Patel	Full Stack Developer	Scrum Master
Meet Patel	Full Stack Developer	Testing
Rutvik Patel	Full Stack Developer	Documents manager
Tirth Patel	Full Stack Developer	Database Manager
Paras Patel	Full Stack Developer	Copywriter and designer

References

[1] "Reasons to rent", *ERA*, 2022. [Online]. Available: https://erarental.org/reasons-to-rent/. [Accessed: 16- Jun- 2022].

[2] free, “Flowchart Maker & Online Diagram Software,” *Diagrams.net*, 2022. [Online]. Available: https://app.diagrams.net/ [Accessed: Jun. 15, 2022].

[3] “Balsamiq. Rapid, Effective and Fun Wireframing Software | Balsamiq,” *Balsamiq.com*, 2022. [Online]. Available: https://balsamiq.com/ [Accessed: Jun. 15, 2022].

[4] “Balsamiq. rapid, effective and fun wireframing software: Balsamiq,” . Rapid, Effective and Fun Wireframing Software. [Online]. Available: https://balsamiq.com/. [Accessed: 04-Jun-2022].

[5] 2022. [Online]. Available: https://www.unitedrentals.com/. [Accessed: 17- Jun- 2022].

[6] “Material UI - overview,” Mui.com. [Online]. Available: https://mui.com/material-ui/getting-started/overview/. [Accessed: 06-Jun-2022].

[7] “Installation,” Mui.com. [Online]. Available: https://v4.mui.com/getting-started/installation/. [Accessed: 07-Jun-2022].

[8] “CSS gradients,” W3schools.com. [Online]. Available: https://www.w3schools.com/css/css3_gradients.asp. [Accessed: 07-Jun-2022].

[9] “Intelligent diagramming,” Lucidchart. [Online]. Available: https://www.lucidchart.com/. [Accessed: Jun 4, 2022].

[10] “Node.Js,” Node.js. [Online]. Available: https://nodejs.org/en/. [Accessed: Jun 4, 2022].

[11] V. Subramanian, “React-Bootstrap,” in Pro MERN Stack, Berkeley, CA: Apress, 2019, pp. 315–376.

[12] “Renting electronic device. renting electronics website, new device rent, terms of use and conditions, gadget rental, test equipment lease. Free Vector,” *Freepik*, Mar. 31, 2021. [Online]. Available: https://www.freepik.com/free-vector/renting-electronic-device-renting-electronics-website-new-device-rent-terms-use-conditions-gadget-rental-test-equipment-lease_13450501.htm#query=equipment%20rental&position=12&from_view=search. [Accessed: Jun. 04, 2022]

[13] “Freepik,” *Freepik*, 2020. [Online]. Available: https://www.freepik.com/popular-vectors. [Accessed: Jun. 05, 2022]

[14] https://www.facebook.com/flaticon, “For rent Icon - 1009805,” *Flaticon*, Jun. 05, 2022. [Online]. Available: https://www.flaticon.com/free-icon/for-rent_1009805. [Accessed: Jun. 05, 2022]

[15] “React,” *Reactjs.org*. [Online]. Available: https://reactjs.org/. [Accessed: 16-Jun-2022].

[16] Tutorial Point, *Html tutorial*. Independently Published, 2020.

[17] “CSS tutorial,” *W3schools.com*. [Online]. Available: https://www.w3schools.com/css/. [Accessed: 16-Jun-2022].

[18] B. Botelho and J. Vaughan, “MongoDB,” *SearchDataManagement*, 28-Aug-2020. [Online]. Available: https://www.techtarget.com/searchdatamanagement/definition/MongoDB. [Accessed: 16-Jun-2022].

[19] IBM Cloud Education, “Rest-apis,” *Ibm.com*, 06-Apr-2021. [Online]. Available: https://www.ibm.com/cloud/learn/rest-apis. [Accessed: 16-Jun-2022].