

SW Engineering CSC648/848 Spring 2022

GatorTrade

Team05

Milestone 2

20 March 2022

Members:

Team Lead: Kishan Patel - kpatel0@sfsu.edu

Front End Lead: Michael Cheung

Front End Member: Joanne Wong

Back End Lead: Faisal Zaheer

Back End Member: Juan Hernandez

Github Master: Ze Lei

History Table:

Date Submitted	Date Revised
20 March 2022	

	2
Executive Summary:	3
List of Data Items and Entities - Expanded As Necessary	4
Functional Requirements - Prioritized	6
Priority 1:	6
Priority 2:	6
Priority 3:	7
UI Storyboards - Low-Fidelity and B/W Diagrams	8
Use Case: Buying a textbook from GatorTrade	8
Use Case: Reviewing the seller after a transaction	11
Use Case: Steve receives a message and completes a transaction	15
Use Case: Managing the GatorTrade application	18
Use Case: Listing a job opening	20
High-Level Architecture and Database Organization Summary	23
Key Risks	24
Project Management	25

1. Executive Summary:

Currently, there is a lack of a dedicated platform for the students, faculty, and staff of San Francisco State University (SFSU) to quickly and securely exchange goods. Students often try to find the cheapest and most efficient way to get the items they need for their classes or for their stay at SFSU. If a student wanted to sell a used textbook for a particular class, it is hard to find a suitable platform since they would not be certain what type of platform other students would use to find used textbooks for sale. Although there are some undedicated places for SFSU members to sell their belongings like Amazon, SFSU Reddit, and SFSU Discord, they have less traffic and exposure since not every SFSU member would use or know about that platform. Students and employees that have hobbies in arts and crafts can also list their handicrafts on this platform to be sold and exchanged with students and fellow coworkers. Having a dedicated platform can also provide a sense of safety for the users, especially since having a place to have secure transactions can allow for accountability to occur.

Our application will be tailored to the students, faculty, and staff at SFSU. Belongings they may not want anymore or think others can benefit from can be listed here. The platform will allow for SFSU students, faculty, and staff to buy, sell, or exchange items with one another. This means that the used textbooks, electronics, and other used or handcrafted goods can find new owners amongst the SFSU community. Users of this application shall be able to search and filter through tags for the goods they are looking for such as technology, textbooks, or crafts. Also, since this application is exclusively for the SF State community, users will get some unique features like being able to search for class materials by class name. It can also be an outlet for artistic students to showcase their skills and sell handmade goods to fellow students and SFSU employees.

Our team is a student start-up composed of students at SFSU. Each member is well-versed in the problems and concerns we address with our application since we have all experienced it firsthand at least once. Therefore, it is safe to say that we know the target audience well because we are in fact part of that same target audience the application strives to appeal to.

2. List of Data Items and Entities - Expanded As Necessary

- a. Users: stores user information such as user password, email, and name.
 - 1. `idUsers`: primary id used in database identification of the user
 - 2. `password`: user-created key used for login
 - 3. `email`: user's SFSU email account; used for login
 - 4. `firstname`: user's first name
 - 5. `lastname`: user's last name

- b. Items: stores item information such as title, price, and description
 - 1. `idItems`: primary id used in database identification of the item
 - 2. `category`: used to reference the `Categories` table
 - 3. `seller`: id that is linked to the user's id
 - 4. `photopath`: path of the photo of the item
 - 5. `description`: description of the item
 - 6. `title`: title of the item
 - 7. `price`: price set for the item by seller
 - 8. `approved`: boolean to check if the post has been approved by admin or not

- c. Reviews: stores review information such as reviewee, rating, and reviewer
 - 1. `idReviews`: primary id used in database identification for review
 - 2. `reviewer`: used the reference the `Users` table for the person making the review
 - 3. `reviewee`: used to reference the `Users` table for the person being reviewed
 - 4. `rating`: value given to the reviewer gave the reviewee
 - 5. `header`: title of the review
 - 6. `body`: context of the review

- d. Messages: stores information of the messages being sent [one way only]

1. `idMessages` : primary id used in database identification of messages
 2. `sender` : used to reference the `Users` table for the person sending the message
 3. `receiver`: used to reference the `Users` table for the person receiving the message
 4. `body`: content of the message
- e. Categories: It will contain the following fields for title: Electronic, Text, Handicraft, Educational_supply, and Food.
1. `idCategories`: primary id used in database identification of categories
 2. `title`: name of the category

3. Functional Requirements - Prioritized

Priority 1:

1. Unregistered Users:

- 1.1** Unregistered users shall be able to register a new account with a valid SFSU email address.
- 1.2** Unregistered users shall be able to access the homepage of the website.
- 1.3** Unregistered users shall be able to browse items on the website.
- 1.4** Unregistered users shall be able to view details of an item, including price, image, item details, etc.
- 1.5** Unregistered users shall be able to search for items on the website.
- 1.6** Unregistered users shall be prompted to create an account (lazy registration).

2. Registered Users:

- 2.1** Registered users inherit all functional requirements from Unregistered users.
- 2.2** Registered users shall be able to login
- 2.3** Registered users shall be able to change their password.
- 2.4** Registered users shall be able to post items to sell on the website.
- 2.5** Registered users shall be able to send messages to other registered users.
[one way only]
- 2.6** Registered users shall be able to logout from their account.

3. Admin:

- 3.1** Admin inherits all functional requirements from Registered users.
- 3.2** Admin shall be required to authorize items posted by registered users.

Priority 2:

2. Registered Users:

- 2.7** Registered users shall be able to exchange items with other registered users.
- 2.8** Registered users shall be able to delete items posted by them only.

2.9: Registered users shall be able to write reviews for other registered users only.

3. Admin:

3.3 Admins shall be able to delete any item from the website.

3.4 Admin shall be able to delete user accounts permanently from the website.

Priority 3:

1. Unregistered Users:

1.7 Unregistered users shall be able to sort and filter search results with specific tags and labels.

2. Registered Users:

2.10 Registered users shall be able to edit their user profile picture.

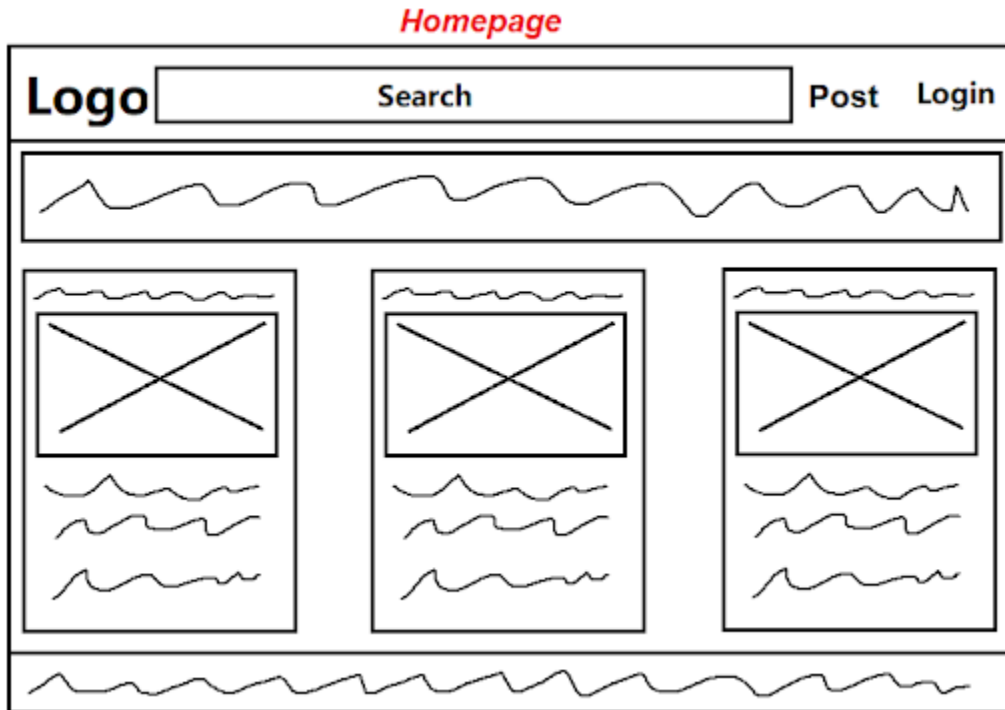
3. Admin:

3.5 Admin shall be able to add or delete categories of items from the website.

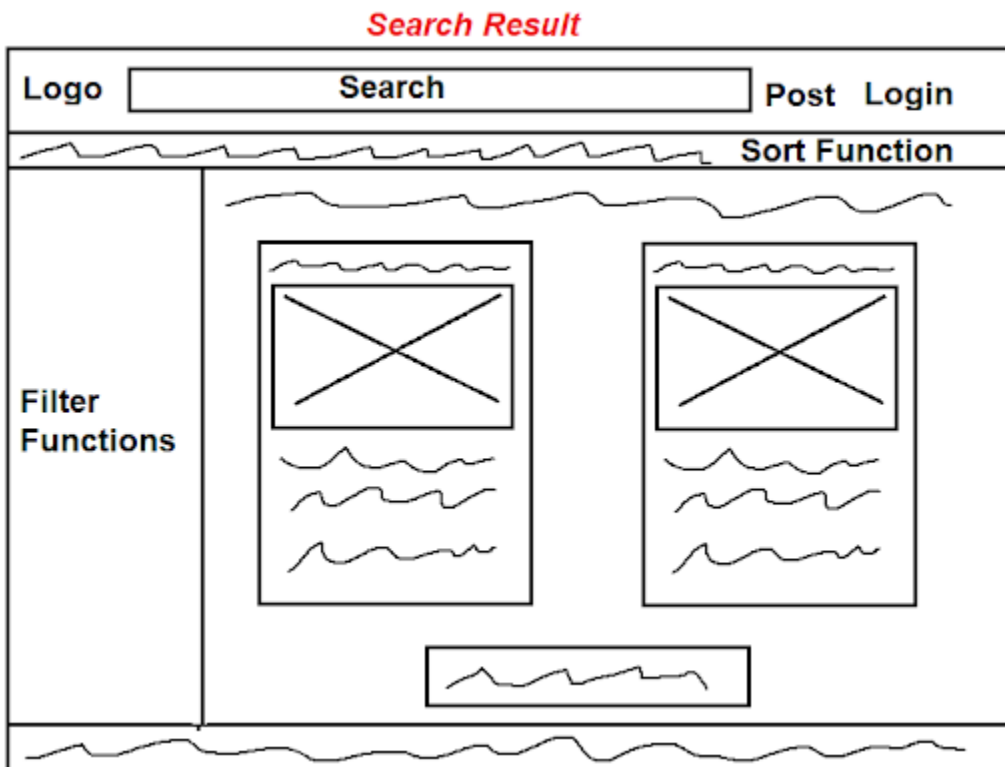
4. UI Storyboards - Low-Fidelity and B/W Diagrams

Use Case: Buying a textbook from GatorTrade

A. Jennifer lands on the homepage and searches for a textbook in the search bar.

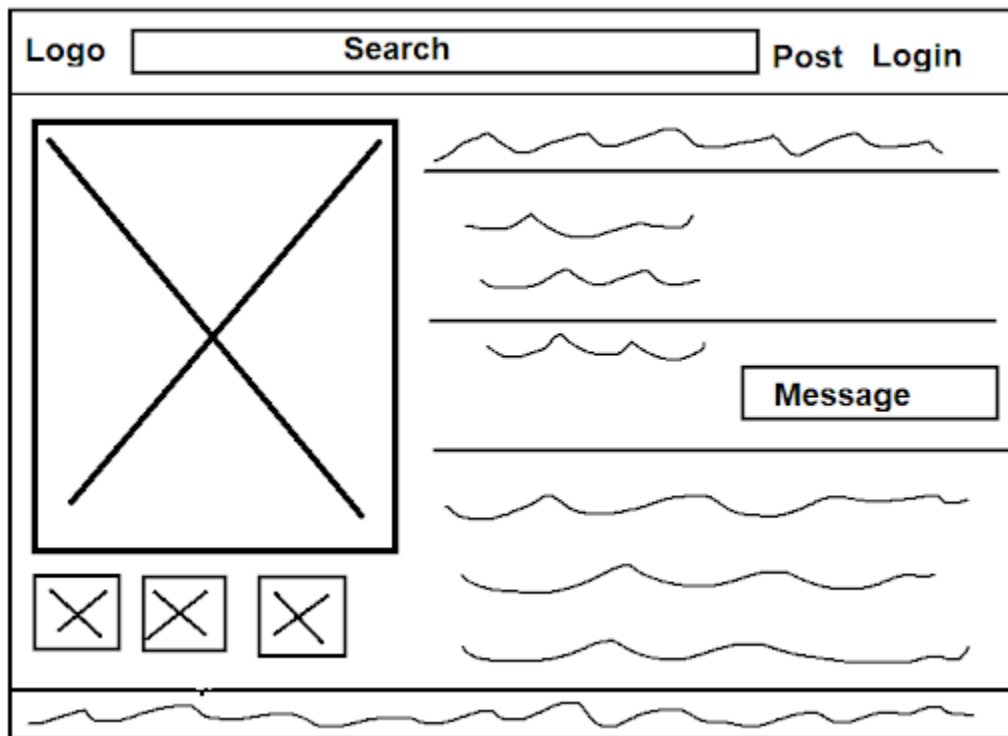


B. Jennifer gets taken to the search results page and clicks on a post



C. Jennifer wants to buy and tries to send a message.

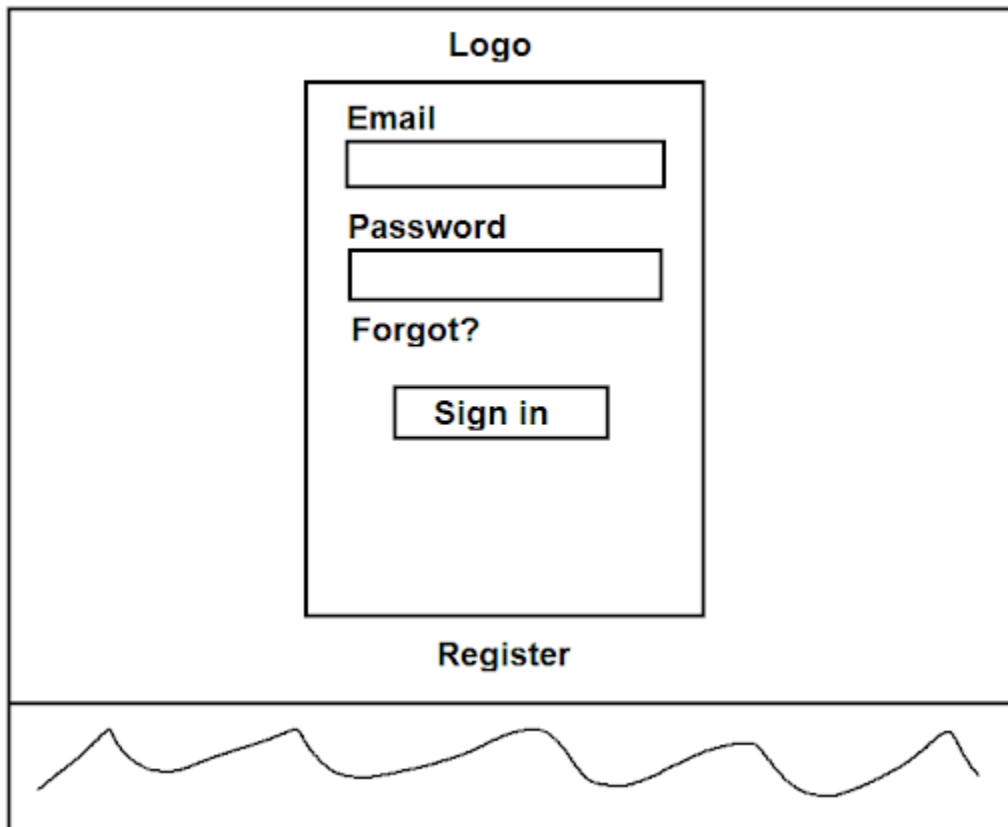
Individual Post Page



The Individual Post Page layout includes a top navigation bar with 'Logo', a 'Search' input field, and links for 'Post' and 'Login'. The main content area is split into two columns. The left column features a large square placeholder with a diagonal 'X' and three smaller square placeholders below it, each also containing an 'X'. The right column contains several horizontal lines representing text, a 'Message' button, and more text lines. A decorative wavy line runs along the bottom of the page.

D. Jennifer cannot send a message unless she is logged in.

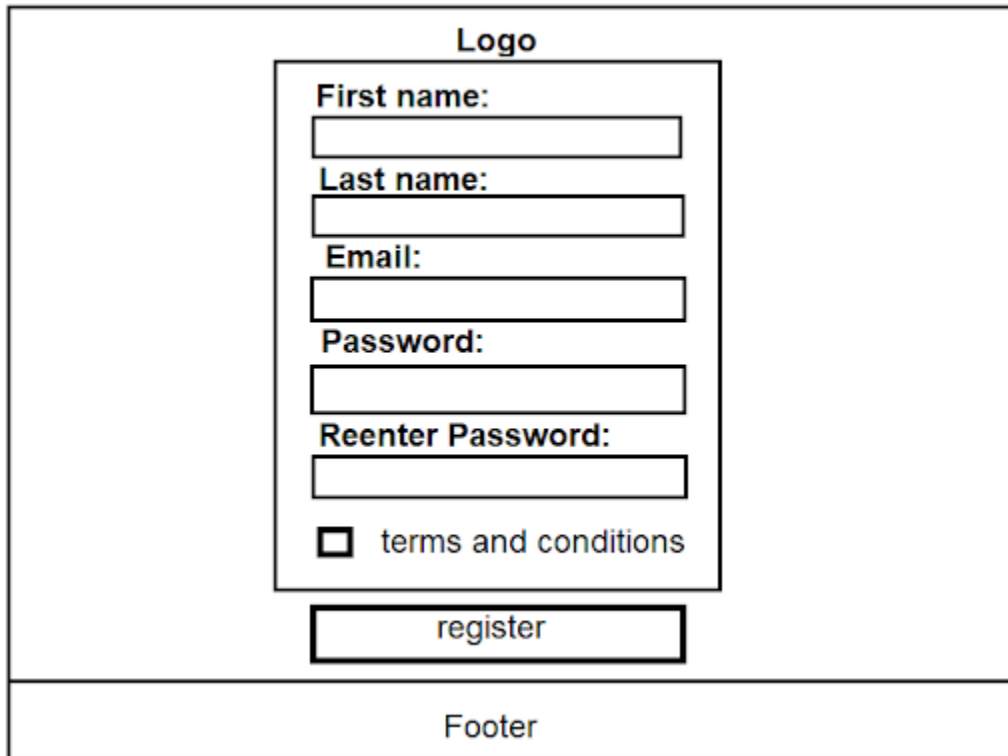
Login Page



The Login Page layout features a central login form with fields for 'Email' and 'Password', a 'Forgot?' link, and a 'Sign in' button. Above the form is a 'Logo' label, and below it is a 'Register' link. A decorative wavy line runs along the bottom of the page.

E. But Jennifer doesn't have an account, so she registers one

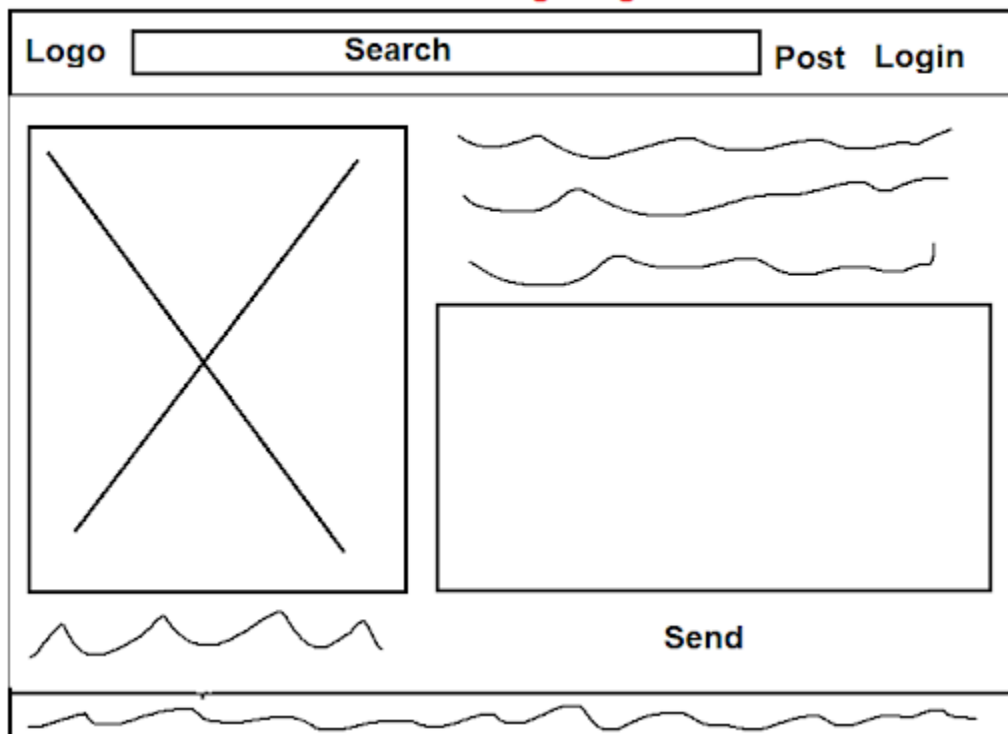
Register page



The register page is a web form with a central registration box. At the top of the page is a 'Logo' label. The registration box contains the following fields and elements: 'First name:' with a text input field, 'Last name:' with a text input field, 'Email:' with a text input field, 'Password:' with a text input field, 'Reenter Password:' with a text input field, and a checkbox labeled 'terms and conditions'. Below the registration box is a 'register' button. At the bottom of the page is a 'Footer' label.

F. After successfully registering for an account, Jennifer gets taken to the message page.

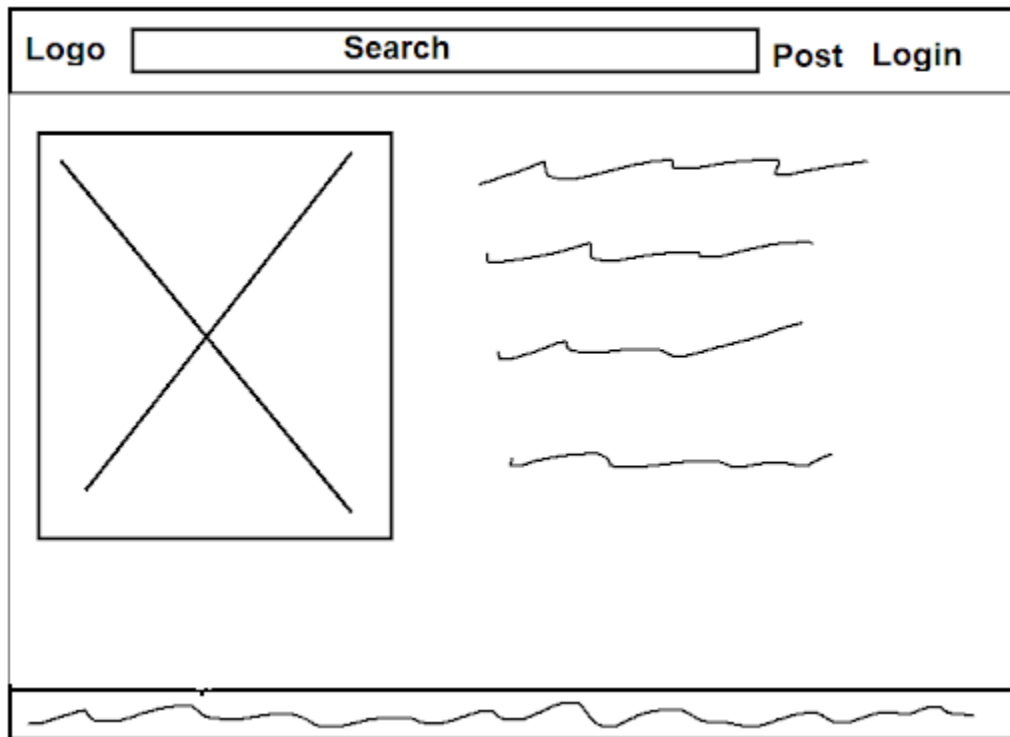
Send Message Page



The Send Message Page is a web form with a header bar containing 'Logo', a 'Search' input field, and 'Post' and 'Login' links. The main content area is divided into two columns. The left column contains a large square placeholder with a diagonal 'X' and a wavy line below it. The right column contains three wavy lines representing text input and a large rectangular placeholder for an image. At the bottom right of the main content area is a 'Send' button. The footer of the page is a wavy line.

G. Jennifer gets a confirmation that her message was sent successfully

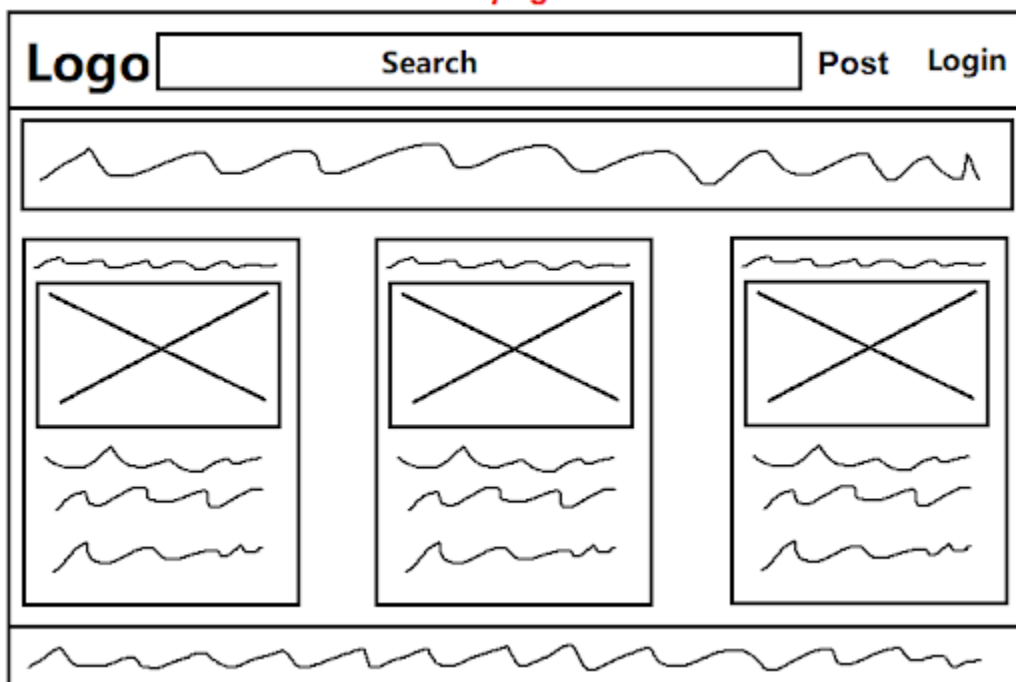
Confirmation Page



Use Case: Reviewing the seller after a transaction

A. Jerry just finished a transaction and the seller asked him to do a review.

Homepage



B. Jerry goes to the login page from the homepage.

Login Page

Logo

Email

Password

Forgot?

Register

C. After successfully logging in, Jerry gets taken to the homepage.

Homepage

Logo

Post **Login**

D. Jerry goes to his account to find his recent messages.

Registered User's Account Page

Logo	<input style="width: 90%;" type="text" value="Search"/>	Post	Login
<div style="border: 1px solid black; width: 100%; height: 20px; margin-bottom: 10px;"></div> <div style="display: flex; justify-content: space-around;"><div style="border: 1px solid black; padding: 5px 10px;">Profile</div><div style="border: 1px solid black; padding: 5px 10px;">Your posts</div><div style="border: 1px solid black; padding: 5px 10px;">Settings</div></div> <div style="border: 1px solid black; padding: 5px 10px; margin-top: 10px;">recent messages</div>			

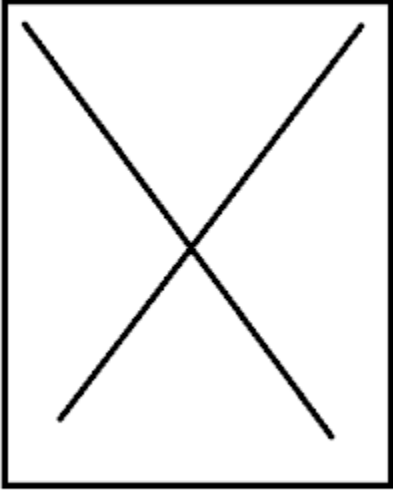
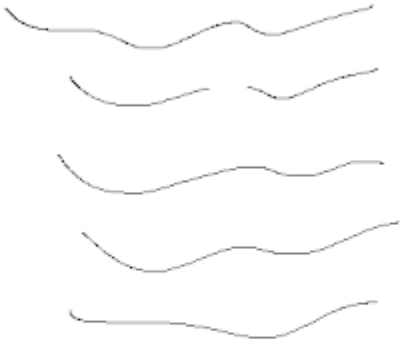
E. Jerry sees the message he sent to the seller he just completed a transaction with.

Recent messages page

Logo	<input style="width: 90%;" type="text" value="Search"/>	Post	Login				
<table border="1" style="width: 100%; border-collapse: collapse;"><thead><tr><th style="width: 50%; padding: 5px;">Sent</th><th style="width: 50%; padding: 5px;">Incoming</th></tr></thead><tbody><tr><td style="padding: 10px; vertical-align: top;"><div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"><input checked="" type="checkbox"/> </div></td><td style="padding: 10px; vertical-align: top;"><div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"><input checked="" type="checkbox"/> </div><div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"><input checked="" type="checkbox"/> </div><div style="border: 1px solid black; padding: 5px;"><input checked="" type="checkbox"/> </div></td></tr></tbody></table>				Sent	Incoming	<div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"><input checked="" type="checkbox"/> </div>	<div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"><input checked="" type="checkbox"/> </div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"><input checked="" type="checkbox"/> </div> <div style="border: 1px solid black; padding: 5px;"><input checked="" type="checkbox"/> </div>
Sent	Incoming						
<div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"><input checked="" type="checkbox"/> </div>	<div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"><input checked="" type="checkbox"/> </div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"><input checked="" type="checkbox"/> </div> <div style="border: 1px solid black; padding: 5px;"><input checked="" type="checkbox"/> </div>						


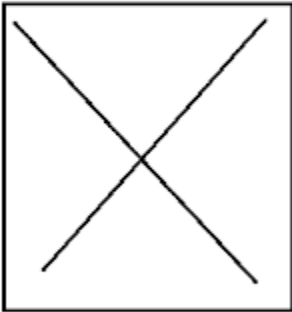

F. Jerry is on the seller's profile page and he clicks the review button.

Registered User's Profile Page

Logo	<input type="text" value="Search"/>	Post	Login
			
		<input type="button" value="review"/>	<input type="button" value="view posts"/>
Footer			

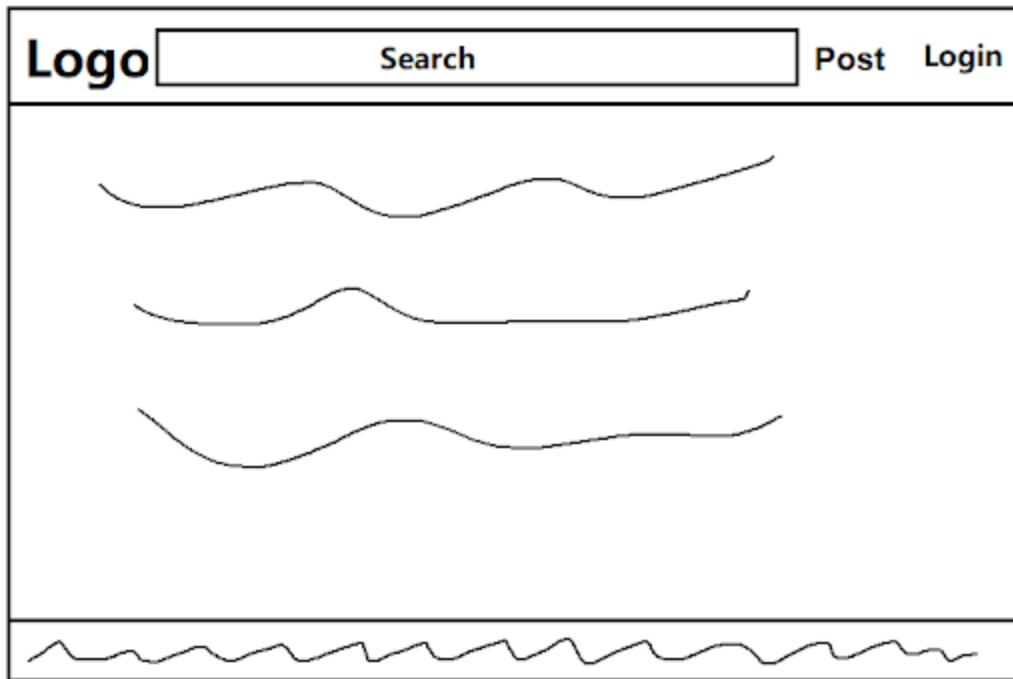
G. Jerry writes his review and clicks submit

Review Page

Logo	<input type="text" value="Search"/>	Post	Login
			
	<input type="text" value=""/>		
<input type="button" value="Submit"/>			
			

H. Jerry gets a thank you page for submitting a review

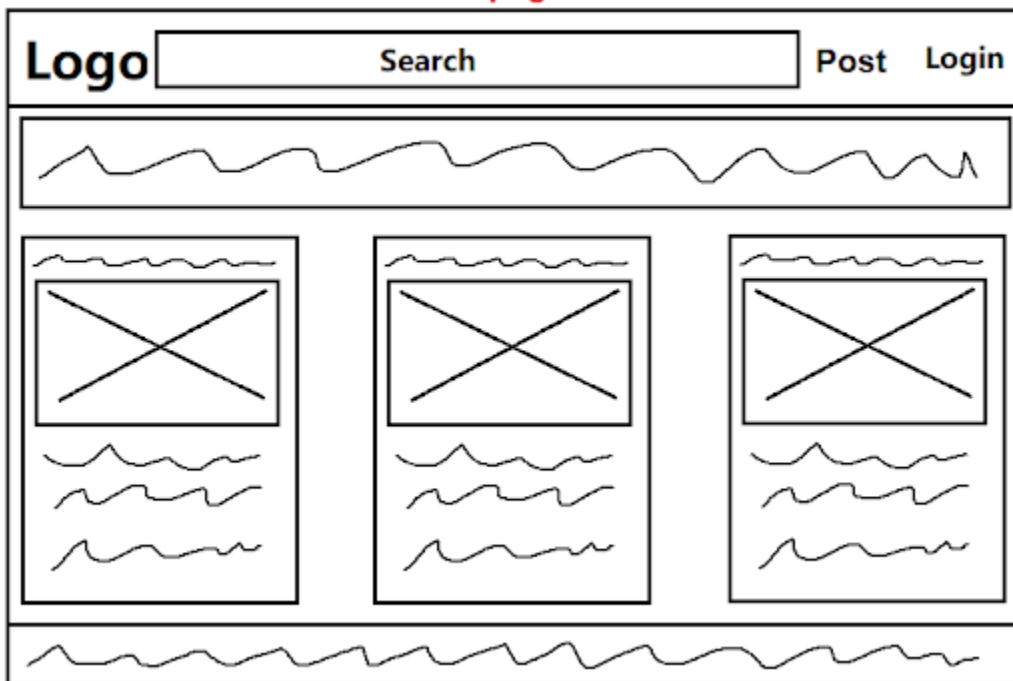
Thank you page



Use Case: Steve receives a message and completes a transaction

A. Steves goes to the homepage to login..

Homepage



B. Steve is taken to the homepage after logging in.

Login Page

The diagram shows a login page layout. At the top center is the text "Logo". Below it is a central rectangular box containing the following elements from top to bottom: the label "Email" above a text input field, the label "Password" above another text input field, the text "Forgot?", and a "Sign in" button. Below this central box is the text "Register". At the very bottom of the page is a decorative wavy line.



C. Steve goes to his account page from the homepage.

Homepage

The diagram shows a homepage layout. At the top is a header bar containing the text "Logo" on the left, a "Search" text input field in the center, and the text "Post Login" on the right. Below the header is a decorative wavy line. The main content area consists of three identical vertical columns. Each column has a square placeholder with a diagonal 'X' at the top, followed by three horizontal wavy lines representing text. At the bottom of the page is another decorative wavy line.


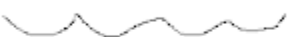




D. Steve is taken to the recent message page after clicking on recent messages

Registered User's Account Page

Logo	<input type="text" value="Search"/>	Post	Login
			
<input type="button" value="Profile"/>	<input type="button" value="Your posts"/>	<input type="button" value="Settings"/>	
<input type="button" value="recent messages"/>			
			

E. Steve checks the most recent incoming message and completes a transaction

Recent messages page

Logo	<input type="text" value="Search"/>	Post	Login
			
Sent		Incoming	
<input type="checkbox"/> 		<input type="checkbox"/> 	
		<input type="checkbox"/> 	
		<input type="checkbox"/> 	
			

Use Case: Managing the GatorTrade application


A. Robert goes to Homepage




Homepage




Logo




Post


Login





B. Robert signs in with his admin account

Login Page


Logo

Email

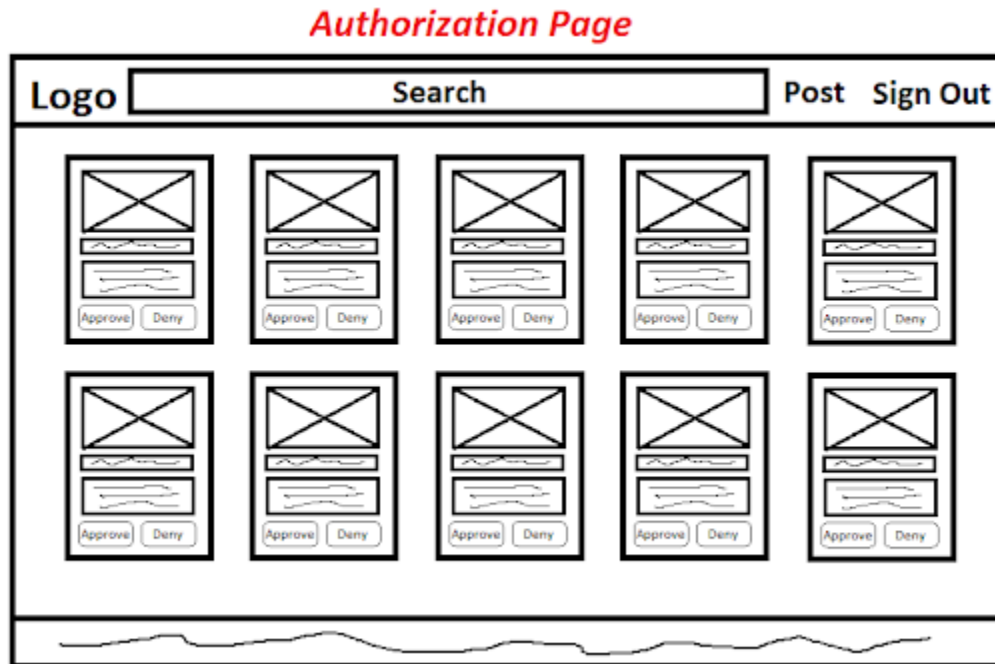
Password

Forgot?

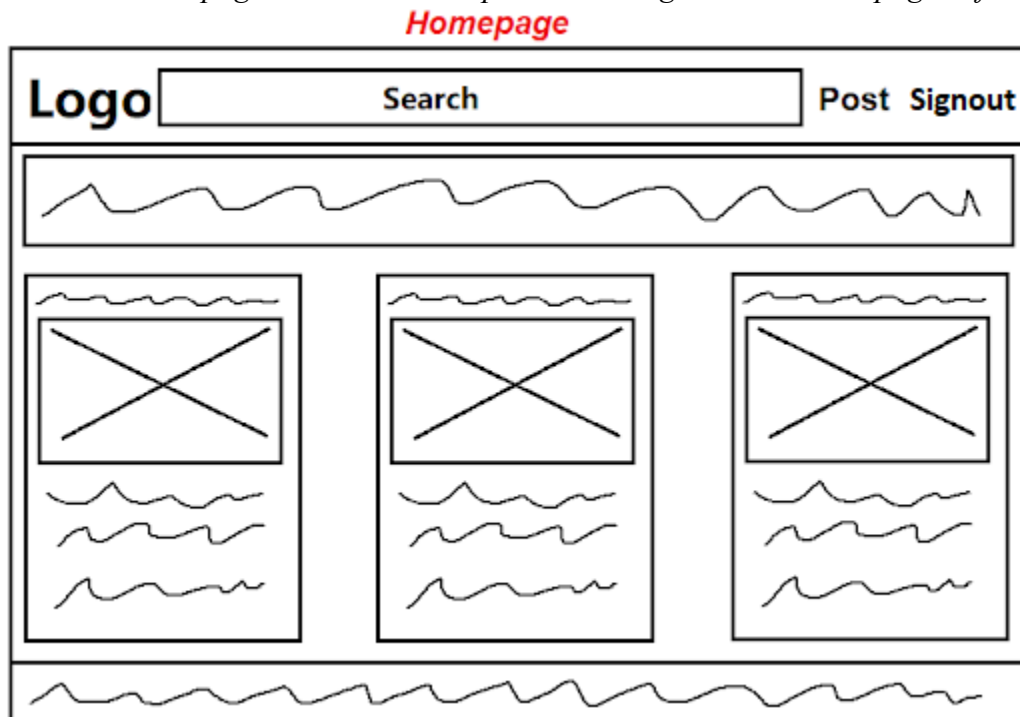
Register



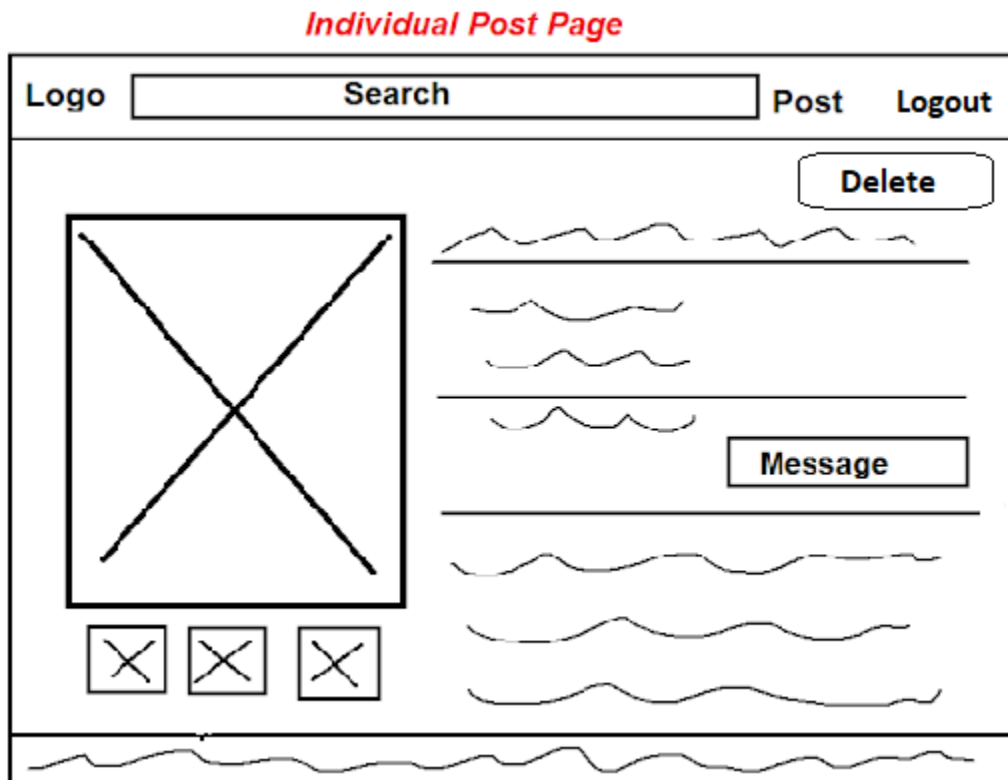
C. Robert goes to review any posts made recently and accidentally approves a post.



D. Since the home page shows the latest posts, Robert goes to the homepage to find the post.

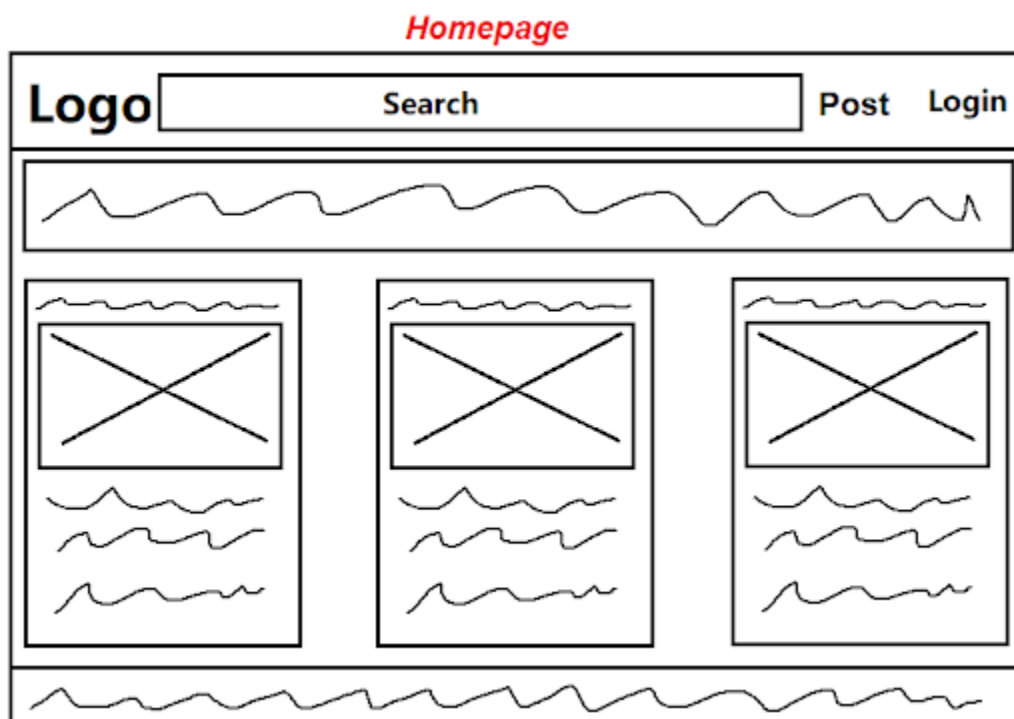


E. Robert clicks on the post and deletes it.



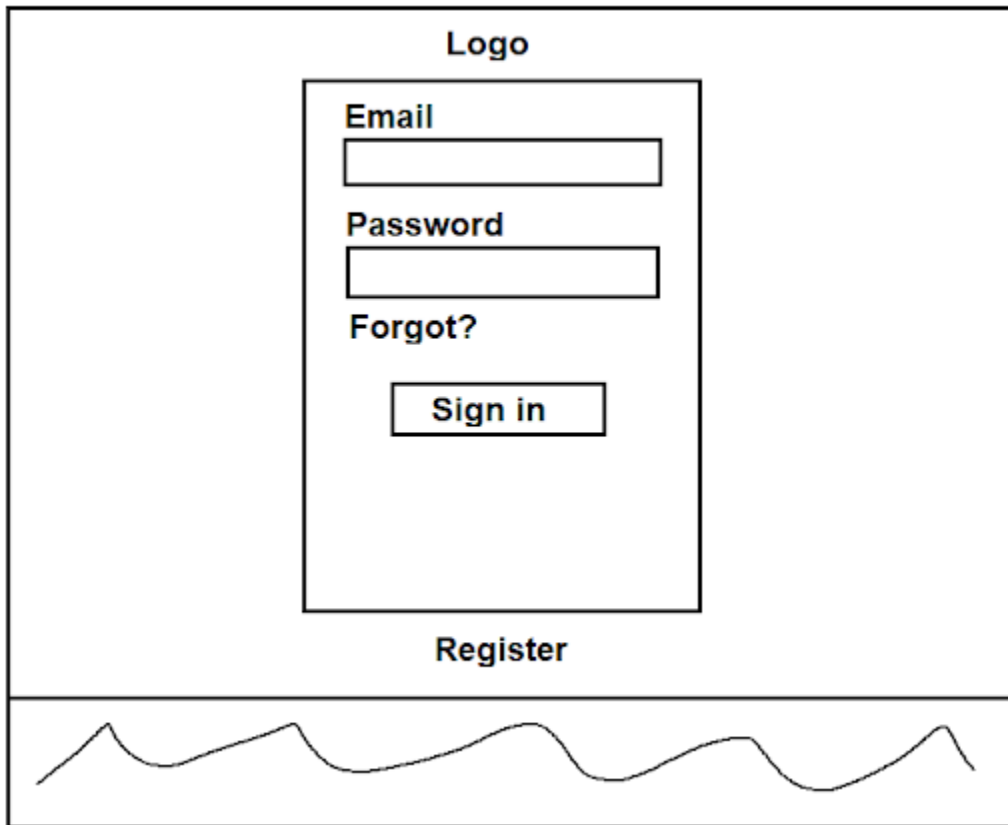
Use Case: Listing a job opening

A. Franklin, an instructor, goes to the Homepage and tries to make a new post.



B. Since he is not logged in, he is taken to the login page.

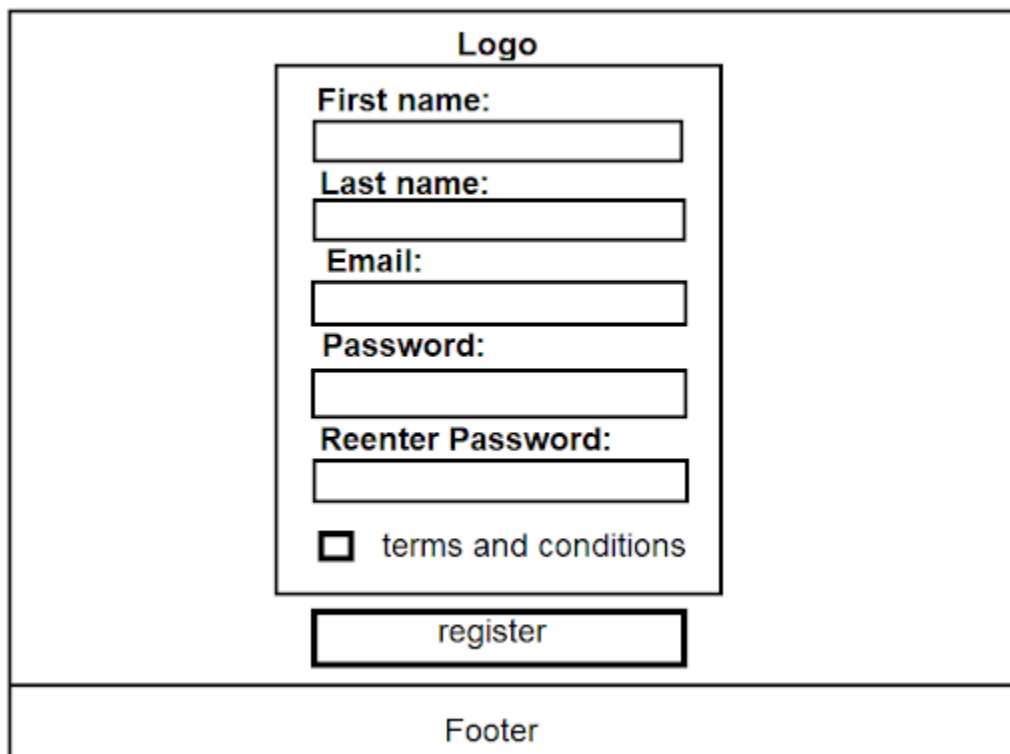
Login Page



The login page form is centered within a large rectangular container. At the top of the container is the word "Logo". Below it is a smaller rectangular box containing the following elements: the label "Email" above a text input field, the label "Password" above another text input field, the text "Forgot?", and a "Sign in" button. Below this inner box is the word "Register". At the bottom of the container is a decorative wavy line.

C. Since Franklin is not a registered user, he registers on the register page

Register page



The register page form is centered within a large rectangular container. At the top of the container is the word "Logo". Below it is a smaller rectangular box containing the following elements: the label "First name:" above a text input field, the label "Last name:" above another text input field, the label "Email:" above a third text input field, the label "Password:" above a fourth text input field, the label "Reenter Password:" above a fifth text input field, and a checkbox followed by the text "terms and conditions". Below this inner box is a "register" button. At the bottom of the container is the word "Footer".

D. After registration, Franklin goes to add a new post and fills out all the information required.

Create Post

Logo

Search

Post

Sign Out

Creating new post

* Title:

* Price:

\$

* Tag:

* Description:

* Photo(s):

Photo function

Submit

Cancel

E. Franklin gets an alert after submitting the form that tells him an admin will review his posting first.

Alert Page

Logo

Search

Post

Sign Out

alert message

okay

5. High-Level Architecture and Database Organization Summary

A. Database Organization:

- a. Users Table:
 - i. idUsers: INT (Primary Key)
 - ii. email: VARCHAR(45)
 - iii. password: VARCHAR(45)
 - iv. firstname: VARCHAR(45)
 - v. lastname: VARCHAR(45)
- b. Items Table:
 - i. idItems: INT (Primary Key)
 - ii. category: INT (Foreign Key)
 - iii. seller: INT (Foreign Key)
 - iv. photopath: VARCHAR(45)
 - v. description: VARCHAR(128)
 - vi. title: VARCHAR(45)
 - vii. price: DOUBLE
 - viii. approved: TINYINT
- c. Categories Table:
 - i. idCategories: INT (Primary Key)
 - ii. title: VARCHAR(45)
- d. Reviews Table:
 - i. idReviews: INT (Primary Key)
 - ii. reviewer: INT (Foreign Key)
 - iii. reviewee: INT (Foreign Key)
 - iv. rating: TINYINT(5)
 - v. header: VARCHAR(45)
 - vi. body: VARCHAR(128)
- e. Messages Table:
 - i. idMessages: INT (Primary Key)
 - ii. sender: INT (Foreign Key)
 - iii. receiver: INT (Foreign Key)
 - iv. body: VARCHAR(128)

B. Media Storage:

Images are stored on the file systems. The webpages have a link to their corresponding images. The images will be in a designated folder. We will downscale all images to save space. We do not have plans to support other forms of media (e.g. video/audio/GPS)

C. Search/filter architecture and implementation:

Search is implemented using the MySQL % Like operator. String comparison will be used to find corresponding items. Search items will be organized in FIFO order, so that old listings can be resolved. Item titles (and potentially descriptions) will be searched. Items can also be filtered using their assigned categories. Categories are selected from a drop down menu.

D. Own API(s): [if any]

None

E. Describe any significant non-trivial algorithm or process if any (like rating, ranking, automatic prioritizing of items etc.):

None

F. Changed SW Tools or Frameworks:

None

6. Key Risks

A. Skill Risks:

- Team needs to learn MVC Architecture.
- Unfamiliarity with AWS

How we will address:

- Team is learning and following the resources and tutorials of MVC and AWS.

B. Schedule Risks: N/A

C. Technical Risks:

- Remote hosting might be an issue temporarily as we have very few free hours left on our AWS account due to an extra instance kept running on the server.

How we will address:

- The issue shall be resolved from the month of April as we have shut down the extra instance.

D. Teamwork Risks:

- Sometimes, there is a communication gap between the frontend and backend teams.

How we will address:

- Team lead has proposed that the backend lead and the frontend lead schedule a weekly meeting where they each discuss the plans and progress of their respective teams.

E. Legal/Content Risks: N/A

7. Project Management

Team members coordinate with each other over Discord and Zoom meetings. The agenda, the work assigned to individual team members and the deadlines are all discussed in the meeting and posted on Discord to keep everyone updated about the same. There are several channels created on the Discord server group which makes it easier to access the particular information for everyone in the team. Once a team member is done with the individual assigned work, he/she updates it on discord to let everyone else know and then also help others if something is left to do before the deadline.

Except for the mandatory zoom meeting after the class, we have our own individual team meeting to discuss the plans for the next few days, assign deliverables and deadlines to all individuals and communicate about any doubts or difficulties. Apart from this general meeting, the backend and frontend teams schedule a short meeting of their own every week. The team lead distributes tasks amongst the frontend and backend team in the general meeting. The frontend and backend leads then distribute tasks to each individual of their respective team in their respective meeting. Also, if the work is interlinked between team members, they schedule internal team meetings on their own to discuss the same.

Team members make sure to complete the tasks assigned and upload the final edited version of it at least a day or two before the deadline so that the document master can just merge all of it, properly format it and submit it on/before time. Once finalized, it is then committed on Github.

As of now, we haven't set up Trello as communication has been pretty smooth over zoom and discord and we didn't want to make things complex. However, we plan to set up Trello after finishing up Milestone 2 as there would be a lot of coding moving forward and hence Trello will simplify communication and project management.