Buy & Sell Platform : Final Report

Pradyot Prakash   Dibyendu Mondal   Sivaprasad S
130050008   130050046   130050085
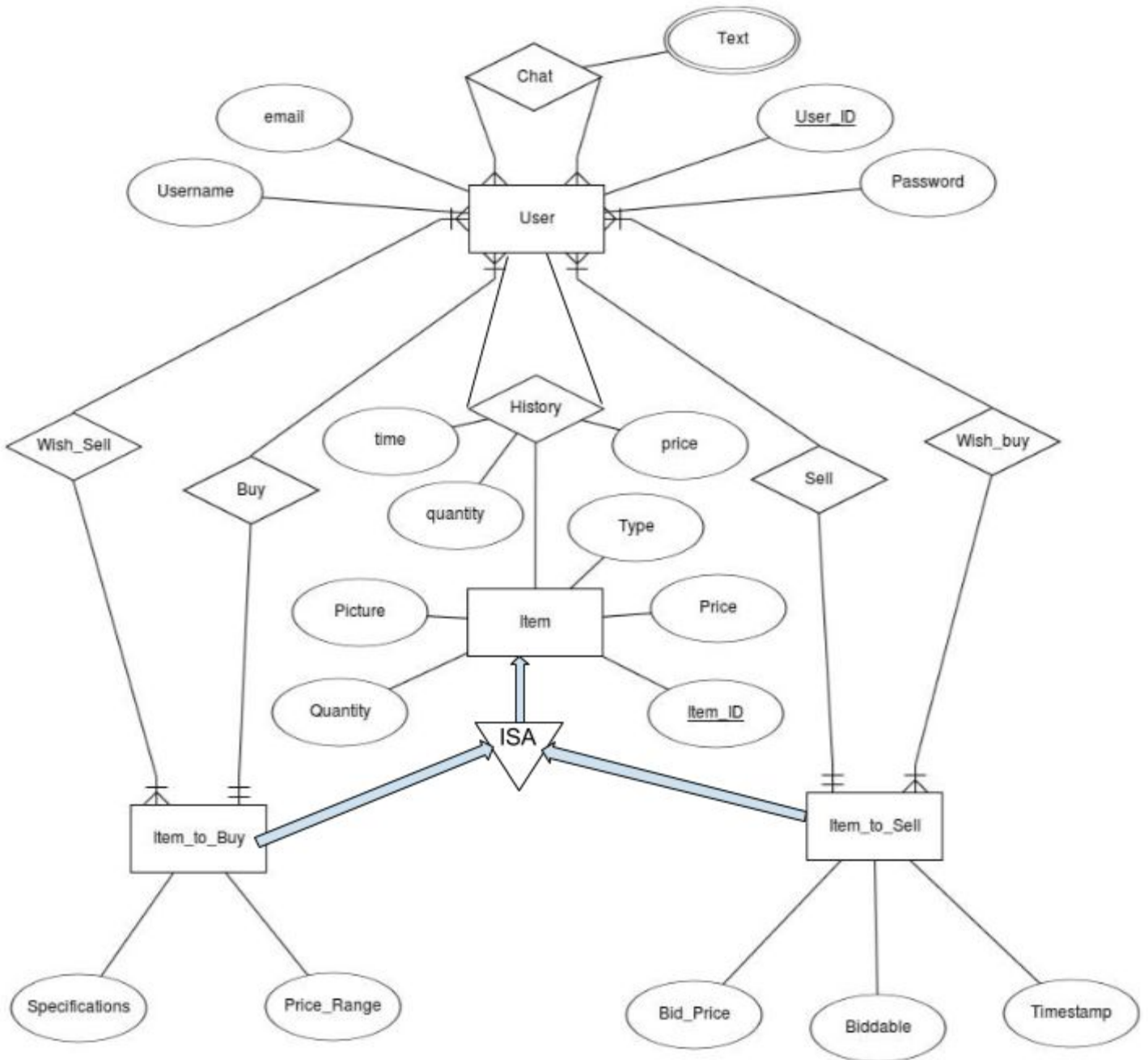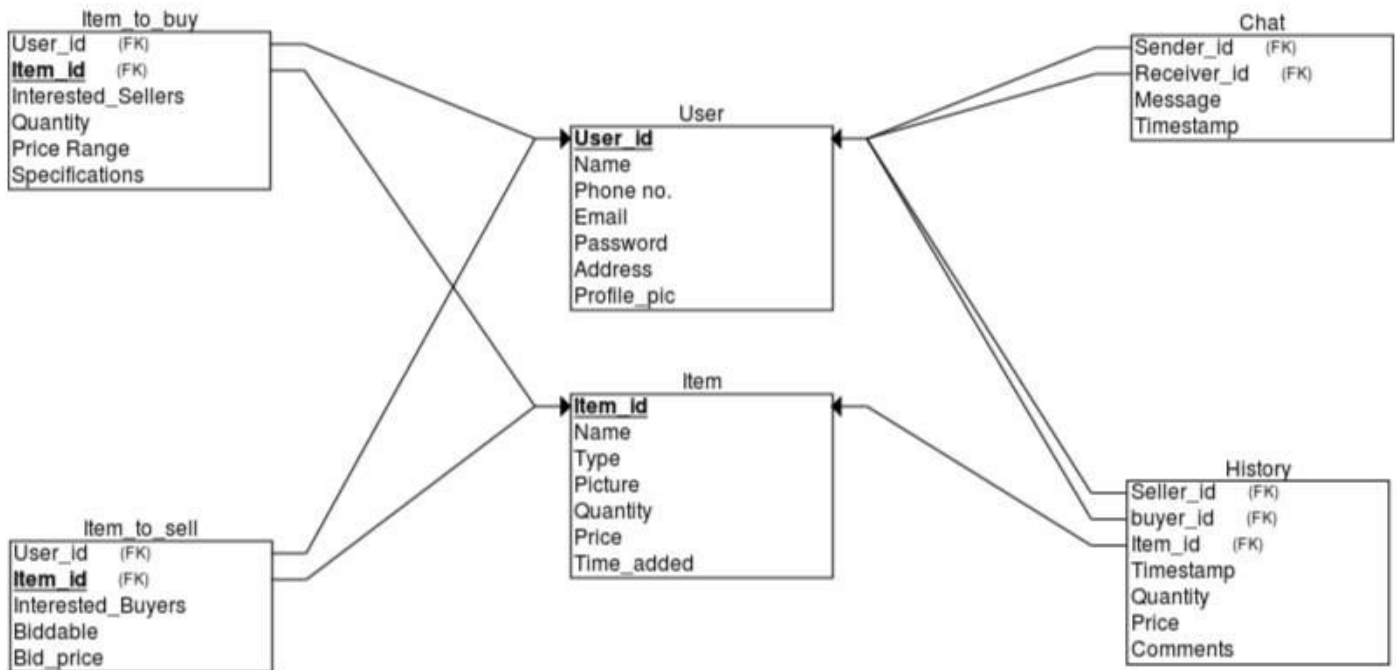
INTRODUCTION

Our project aims to design a web application which would mimic a **Buy and Sell** platform for an organization (eg - IIT Bombay) :

1. People would be able to create their **profiles** with their basic contact information
2. People would be able to **sell/rent** items through the app
3. Each person would have two roles of **buyers and sellers**. As a seller, they can add the things they own and want to put up for sale
4. They would be able to **add the items** they own and want to sell/rent them temporarily
5. There would be a **notification page** where people can post information about the latest item they have added to their selling list
6. People would also be able to post the list of items they need and the sellers can contact them individually
7. There would be a **search bar** functionality where the prospective buyers will be able to search for the item they seek
8. Alternatively, they would be able to look up the profiles of people and select from the items they are interested in
9. This would also have a **chat/messaging** feature. Probably buyers can negotiate the prices and finalize a final price for the item
10. The application would also have an **accounting feature** which would track that which items have been sold and bought
11. We have created an admin interface with permissions to monitor entries in the portal

ER Diagram

Database Schema

Item_to_buy
User_id (FK)
Item_id (FK)
Interested_Sellers
Quantity
Price Range
Specifications

Chat
Sender_id (FK)
Receiver_id (FK)
Message
Timestamp

User
User_id
Name
Phone no.
Email
Password
Address
Profile_pic

Item
Item_id
Name
Type
Picture
Quantity
Price
Time_added

History
Seller_id (FK)
buyer_id (FK)
Item_id (FK)
Timestamp
Quantity
Price
Comments

Item_to_sell
User_id (FK)
Item_id (FK)
Interested_Buyers
Biddable
Bid_price

Approach Used

In order to achieve a basic Buy and Sell platform there are mainly two entities - **User** and **Item**. The item could be added by a user for sale or to show his interest in buying. There are two tables corresponding to **Item_to_sell** and **Item_to_buy** each of them inheriting from the **Item** table.

The first table stores the listings put up by a seller. In our model, an interested buyer would be able to express his/her interest in an item. All the buyers would be able to do this, and at the end the owner would be able to sell the item to whoever person he wishes to. These tables should also have a foreign key referencing the user who has added the item. The latter stores the listings by a user who is interested in buying it.

The relations in the ER diagram : **Buy** : Between the user and the item he is interested in buying, **Sell** : that between the user and the item he has put on sale, **Wish_Buy** : the relation between the users who are interested in buying an item put up in the item_to_sell table, **Wish_Sell** : the relation between the users who are interested in selling an item put up in the item_to_buy table.

They should also store the list of **user_ids** of interested buyers / sellers. The other tables include **Chat** to store the messages exchanged between two users and a **history** table to keep track of the transactions made by the user.

Modules

Classes:
1. Access Database - This class implements the functionality that involves select queries from the database, like getting the list of items available on sale, that are needed by users etc.
2. Update Database - This class implements the functionality that involves updating the database like maintaining the list of items, history, messages etc.
3. Pair - A helper Java class for creating pair of objects
4. Register - A class that implements the function for adding the users to the database
5. Session - A class that helps in identifying the currently logged in user

JSP Files:
1. index.jsp - Login page for the user
2. register.jsp - Sign up page for a new user where he fills his personal details and can create an account
3. home.jsp - The Basic landing page of the user, with links to all the functionality available to him
4. add_listing.jsp - Where he can add an item to sell
5. add_wishlist.jsp - Where he can add an item that he is interested in buying
6. messages.jsp - The Chatting interface, with previous messages and list of users
7. history.jsp - Page where all the previous transactions of a user is available
8. see_listings.jsp - Where he can see the items on sale

Indexes:

A database index is a data structure that improves the speed of data retrieval operations on a database table at the cost of additional writes and storage space to maintain the index data structure. Indexes are used to quickly locate data without having to search every row in a database table every time a database table is accessed. Indexes can be created using one or more columns of a database table, providing the basis for both rapid random lookups and efficient access of ordered records.
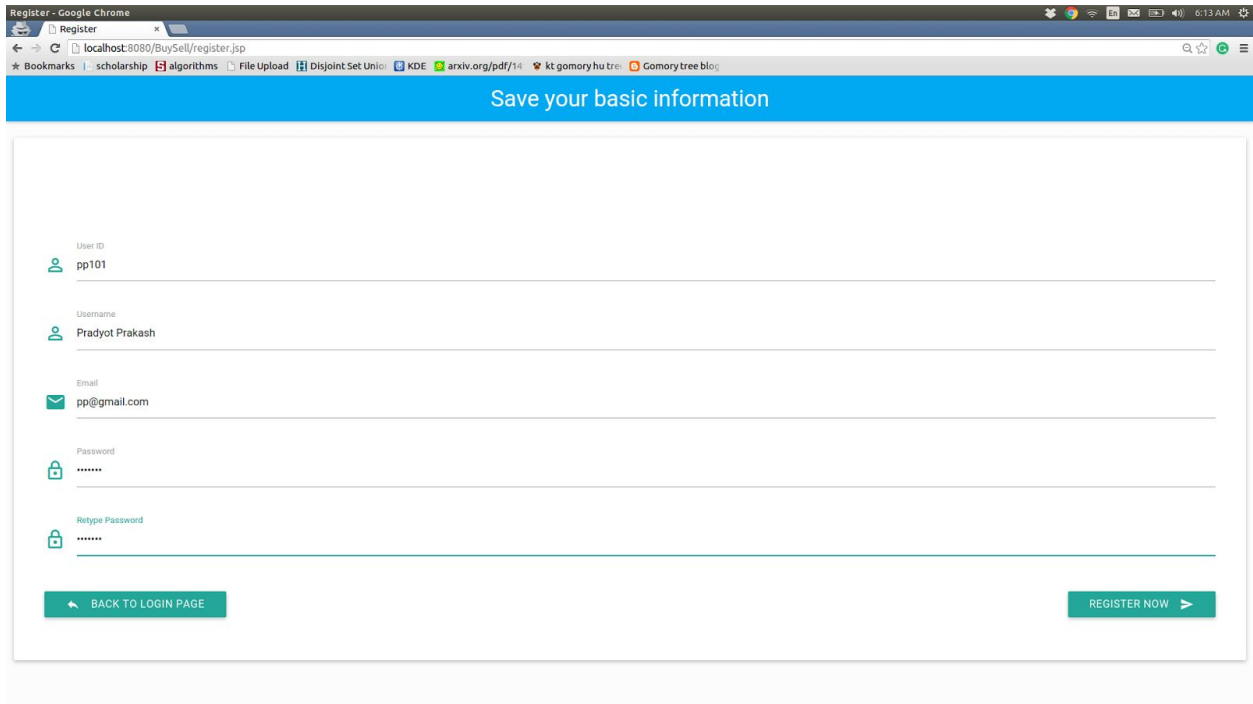
We would be providing indexes for **Item table** on **Type**, **Name** for filtration queries based on these attributes. We created these indexes that provided quick lookups and efficient access of ordered records in the items table.

Tools used

www.erdplus.com for creating ER digram and Database schema.
Eclipse as an IDE

Interface
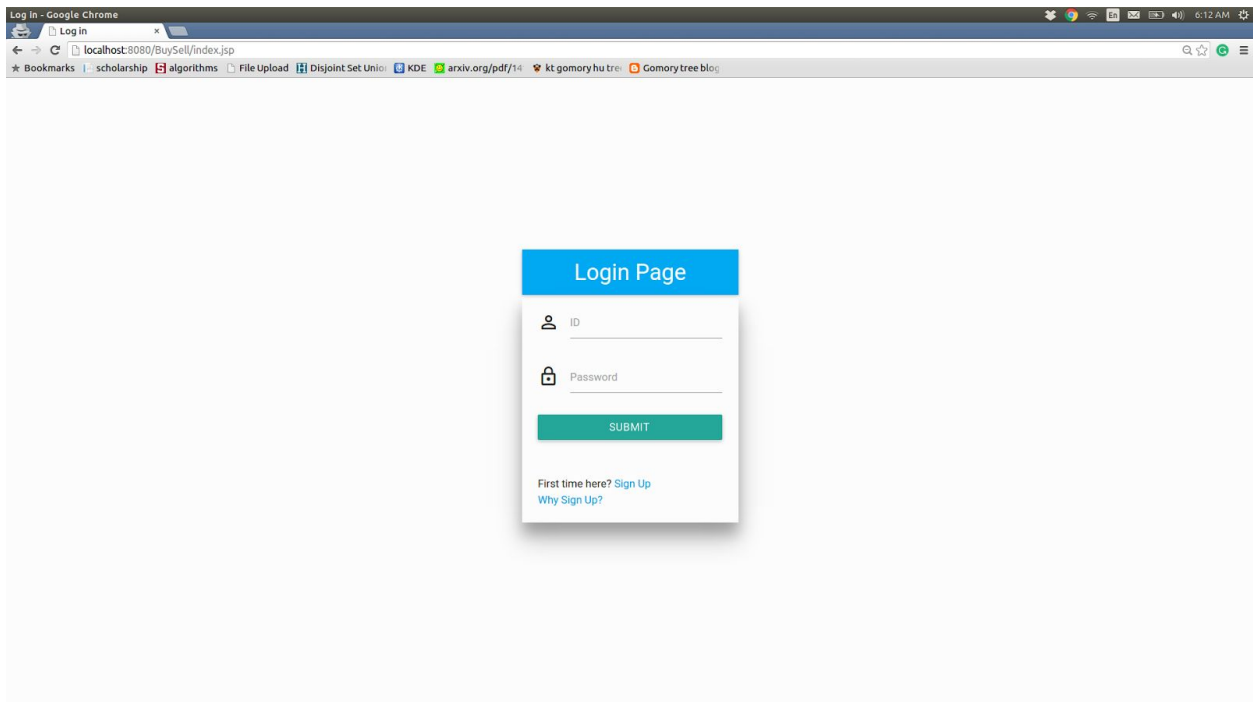
You can register as a user on the portal :

You can login to the portal with verified userid and password:

Each user has a profile on the portal
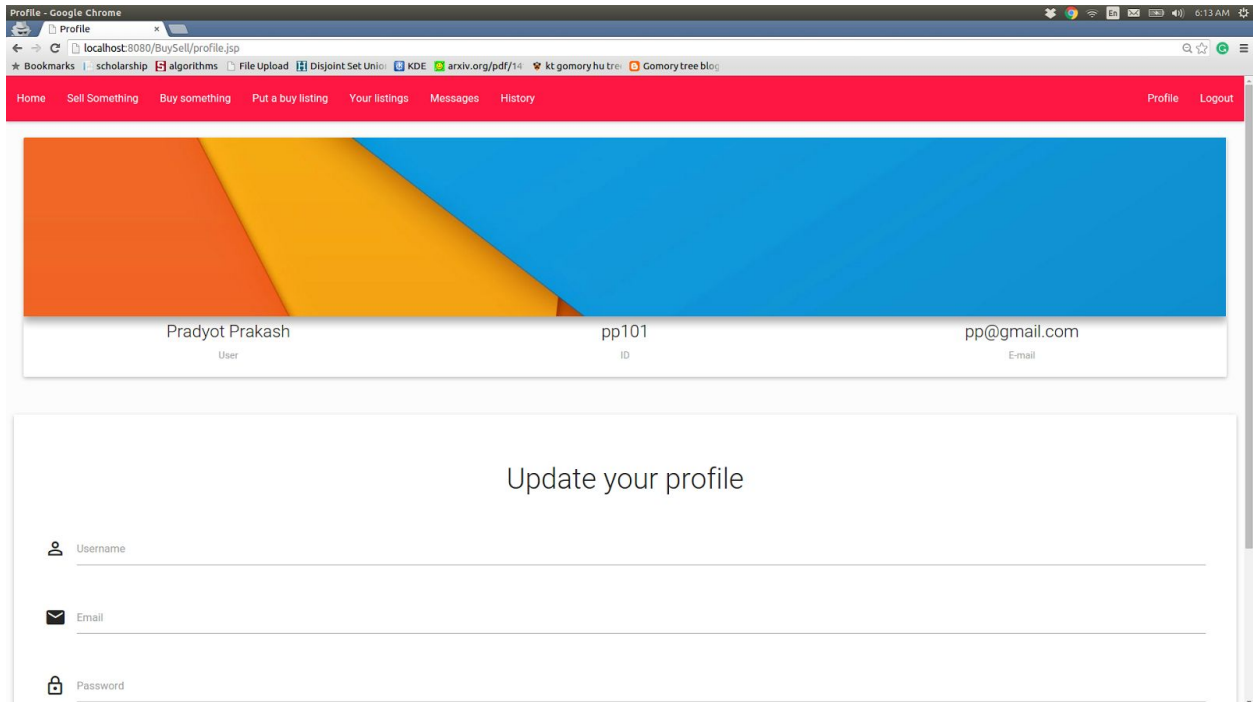
The home page has the posts by other users looking for items in urgency

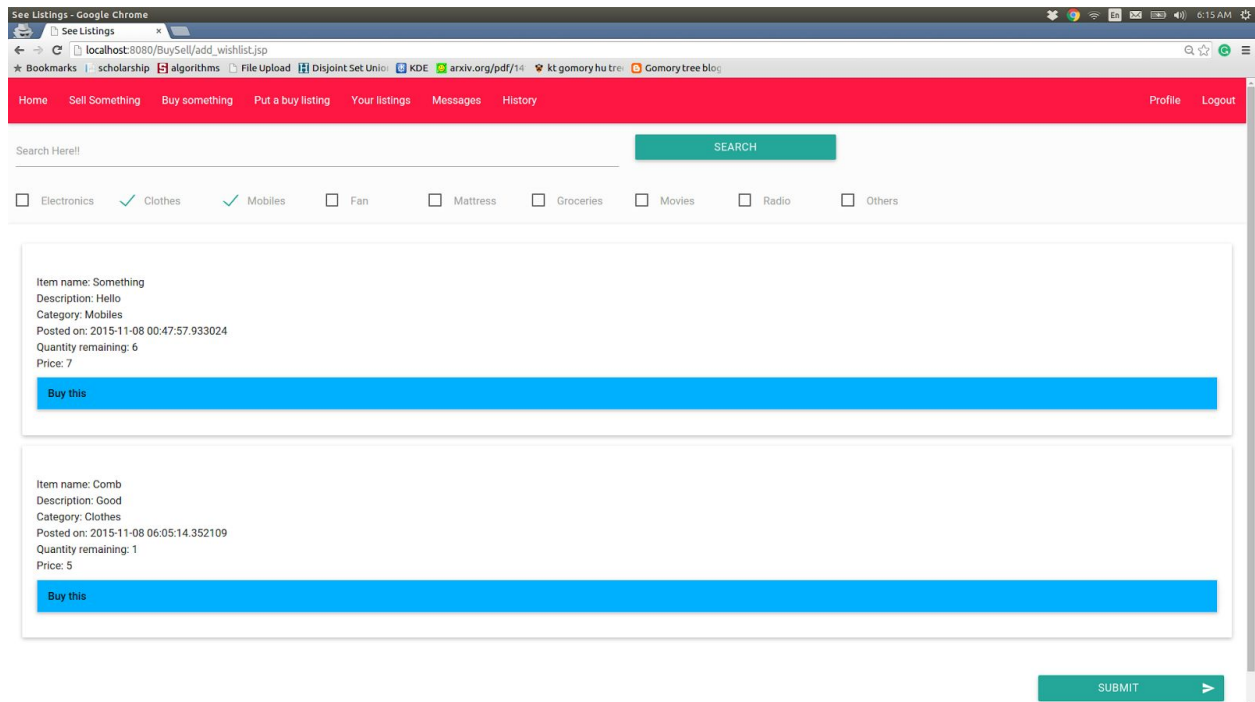Welcome Pradyot Prakash

Recent demands:

Sl. No.	Posted by	Item name	Quantity demanded	Price range	Category	Description	Comments
1	qw	lappy	10	1000-10000	elec	hugga	good
2	pp	laptop	10	9-700	yt	good	tyt
3	pp	t	1	45-76	ere	q	p
4	qw	Handglider	1	10000-20000	Sports	Same as above	Meh!
5	pp	weed	10	10-100	Others	good weed	i wanna get high!!!
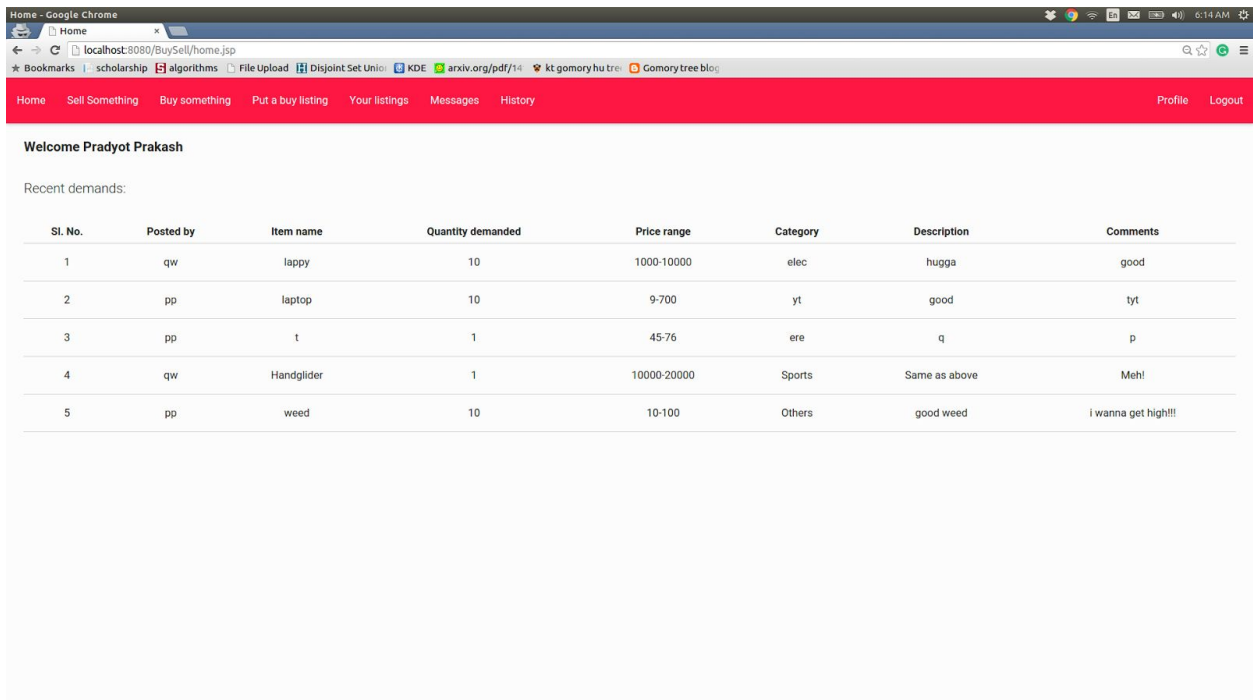
The portal offers you a form to sell things that you own

You can see the items available on sale and buy them if interested

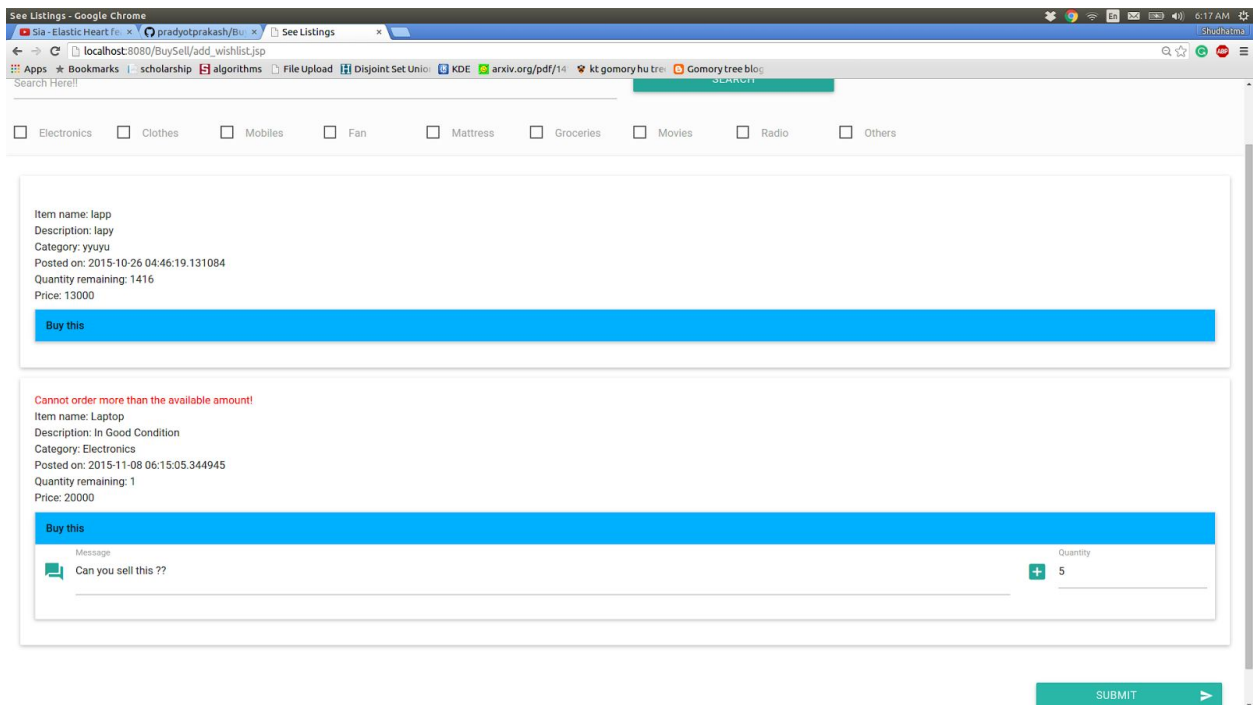The See Listing page offers you various filtration options based on category
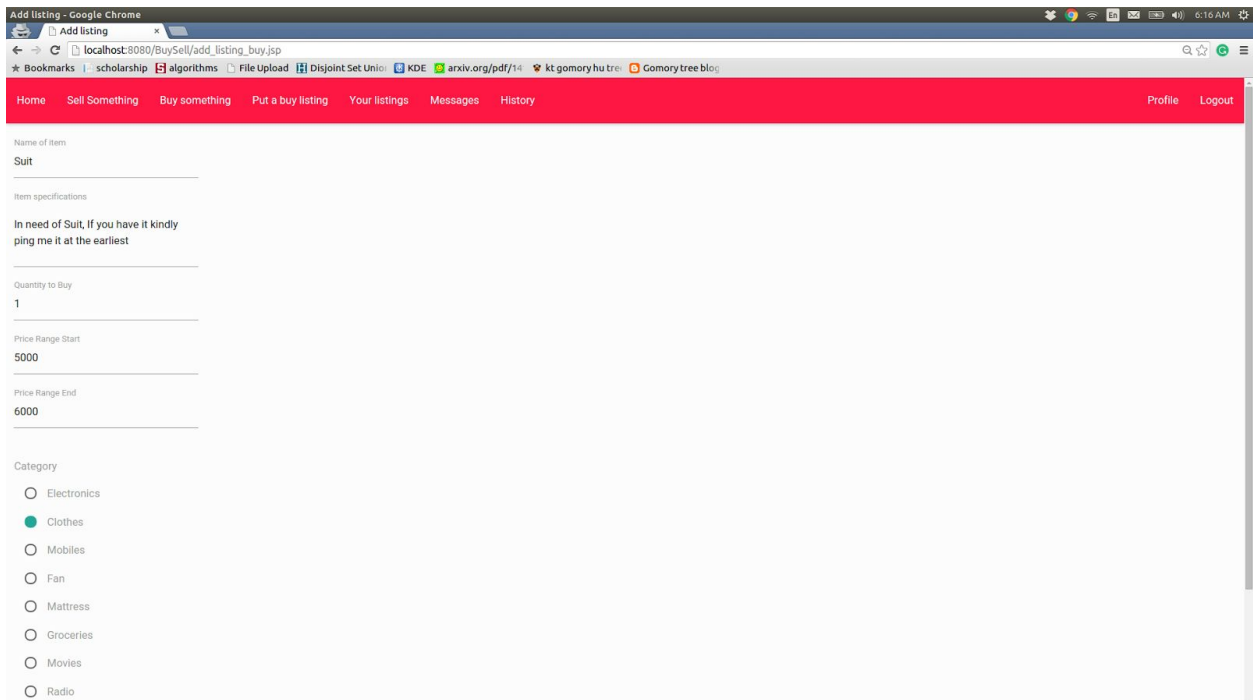

Also offers you a search bar
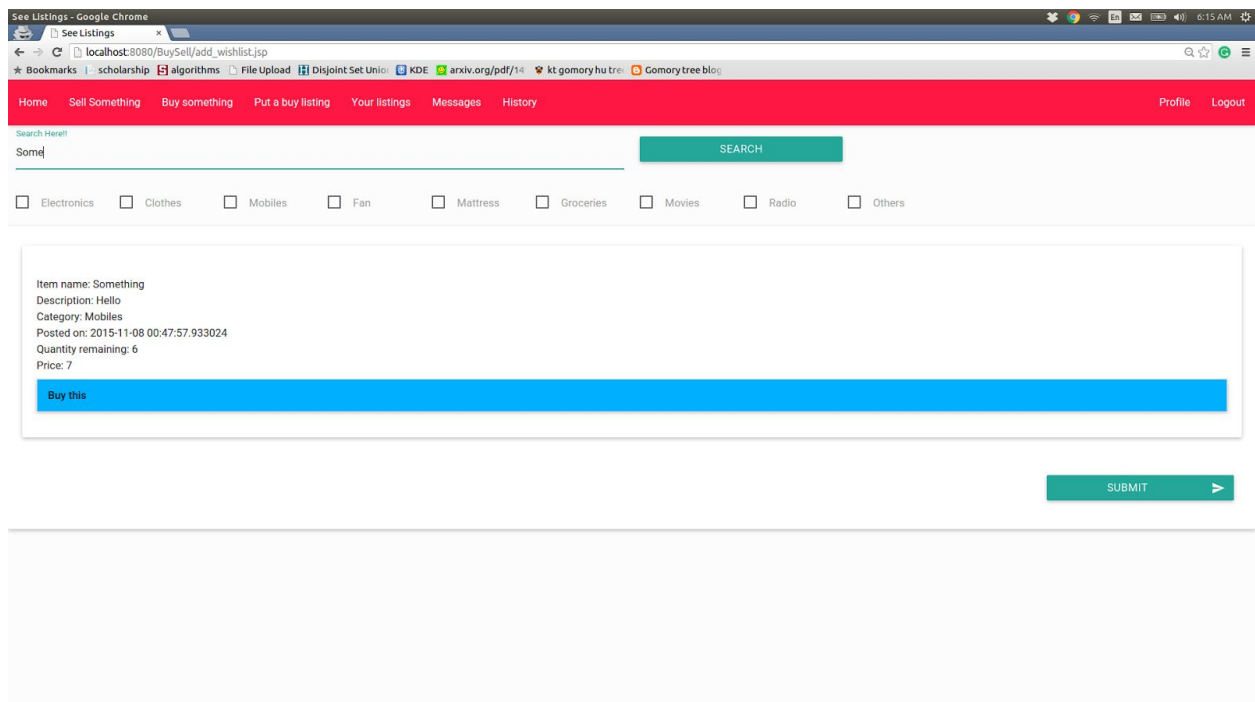
The Buy option ensures basic checks on constraints

You can also post any urgent requirements that you have which will be seen on other's home page

You can see status of your listings

If any user has demanded an item, that will be visible on this page

You may sell this to him, that removes the item from your active listings

localhost:8080/BuySell/see_listings.jsp

★ Bookmarks | scholarship ⊞ algorithms ◻ File Upload ⊞ Disjoint Set Unio ◻ KDE ◻ arxiv.org/pdf/14 ✿ kt gomory hu tre ◻ Gomory tree blog

Home   Sell Something   Buy something   Put a buy listing   Your listings   Messages   History   Profile   Logout

You are not selling anything now!

Things you wish to buy and are not already on this website:

Sl. No.	Item name	Quantity demanded	Price range	Category	Description	Comments
1	Suit	1	5000-4000	Clothes	In need of Suit, If you have it kindly ping me it at the earliest	

This adds it your history page

localhost:8080/BuySell/history.jsp

★ Bookmarks | scholarship ⊞ algorithms ◻ File Upload ⊞ Disjoint Set Unio ◻ KDE ◻ arxiv.org/pdf/14 ✿ kt gomory hu tre ◻ Gomory tree blog

Home   Sell Something   Buy something   Put a buy listing   Your listings   Messages   History   Profile   Logout

Your transaction history as a seller:

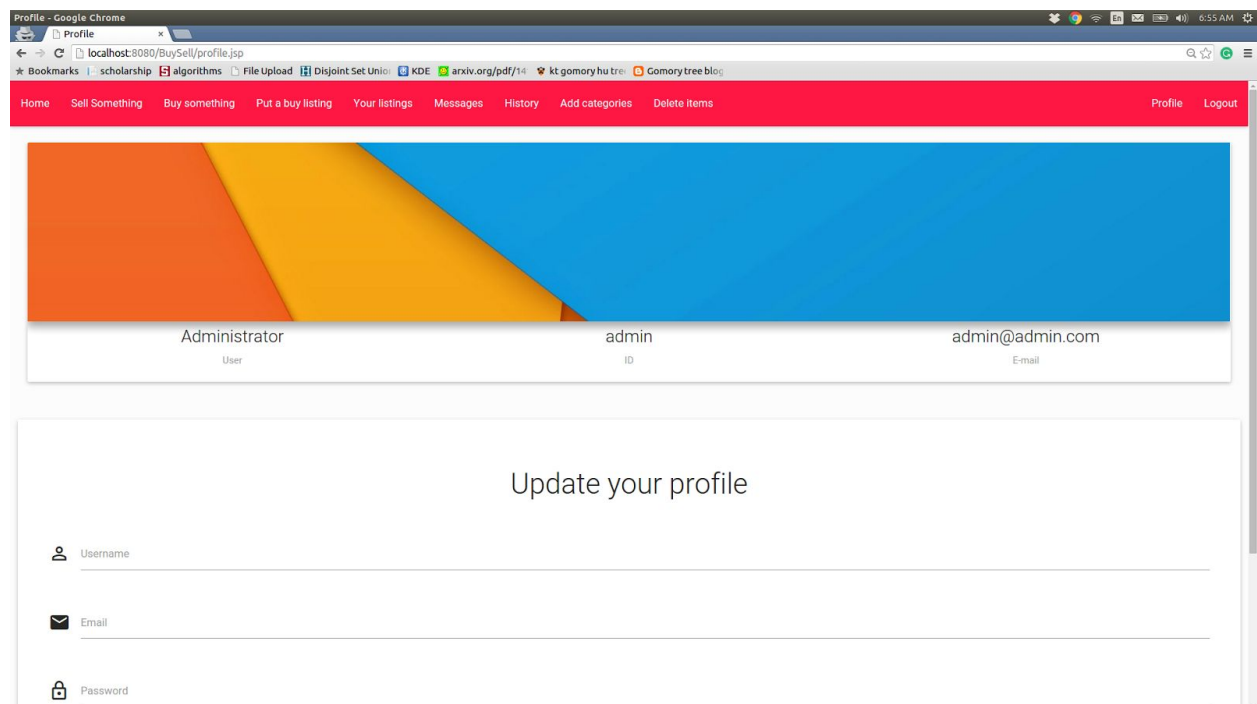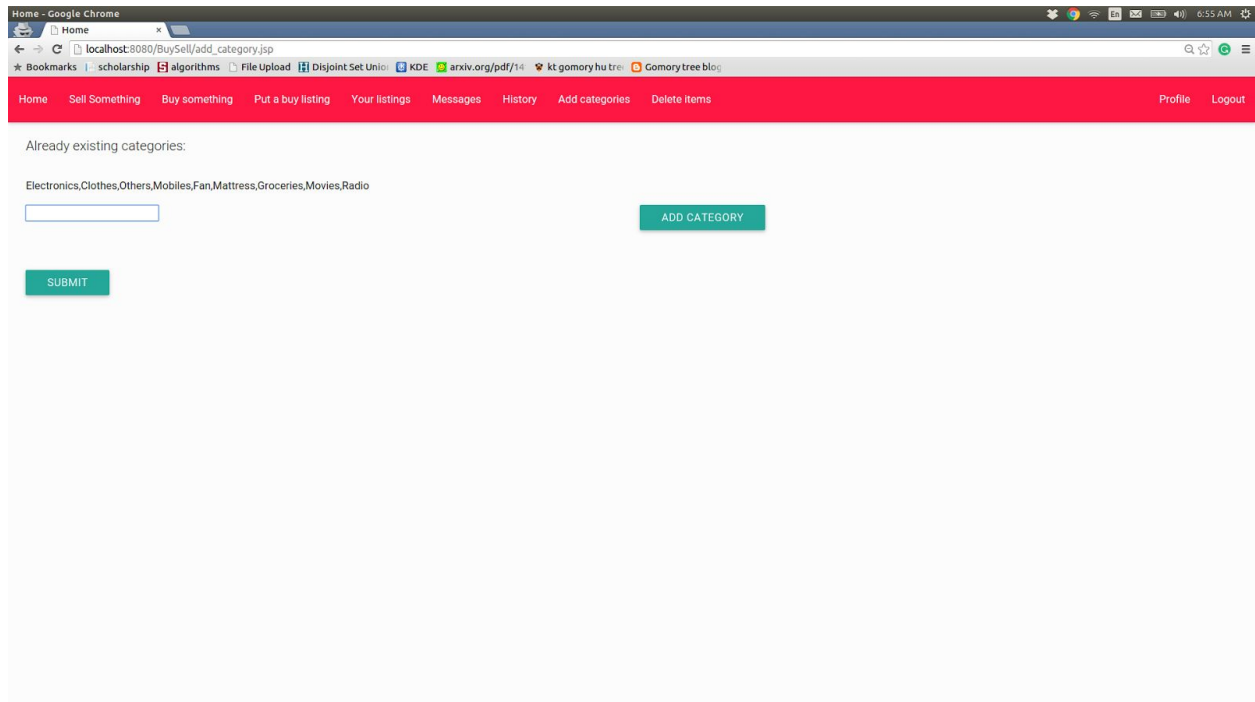| Sl. No.	Item ID	Buyer	Price	Quantity	Time	Comments
1	49	siva	20000.0	1	2015-11-08 06:18:44.276115	null

Your transaction history as a buyer:

No records!

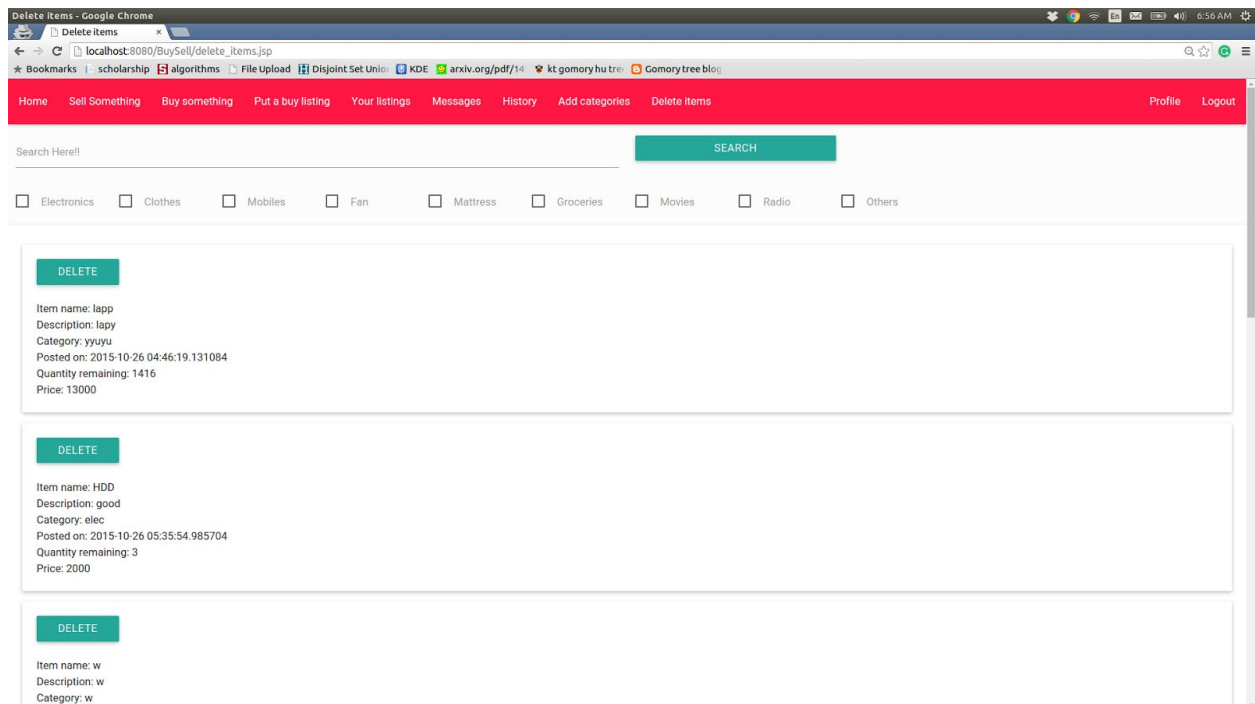You also have a chat option where you may chat with any other user on the portal
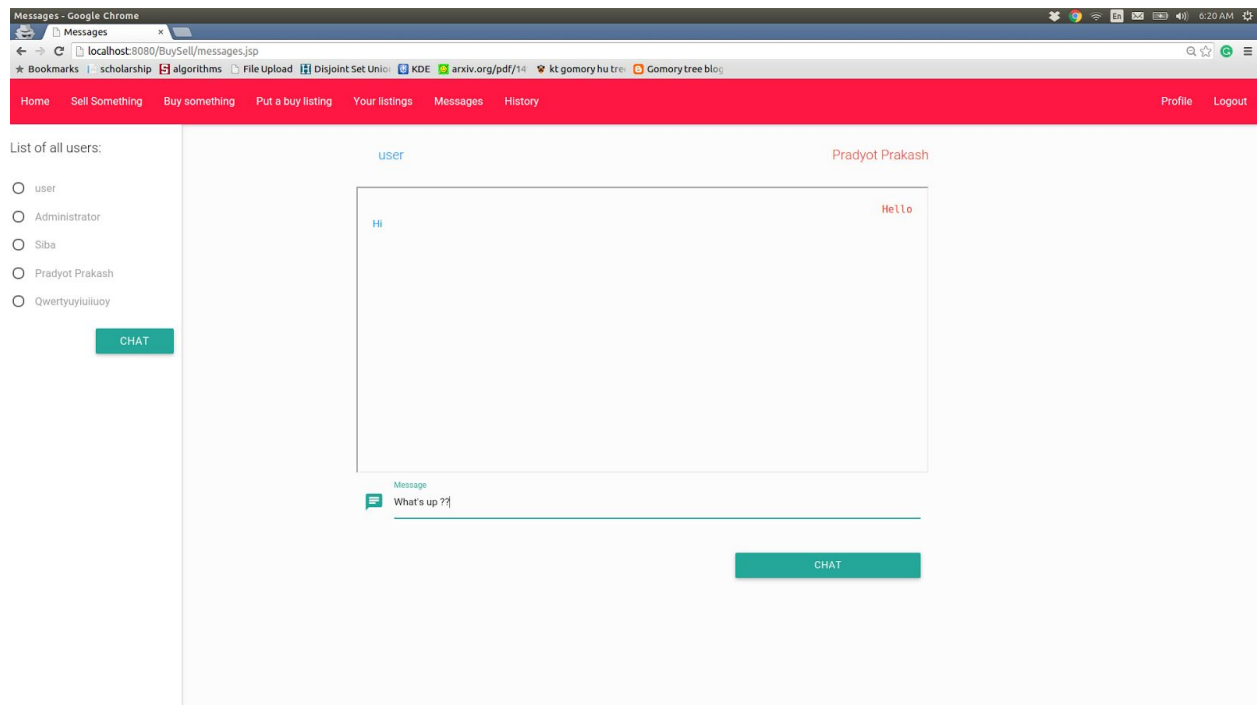
There is an admin interface available

The admin has the power to add categories

The admin can also monitor the items on sale in the portal

References

1. demo.geekslabs.com/materialize/v2.1/layout03/index.html - for css and js
2. www.w3schools.com - for HTML & JavaScript syntax
3. http://stackoverflow.com/ - for correcting some bugs