

Warehouse Management System

By

Katie Nguyen, Jhanvi Lotwala, Alan Park

December 11, 2020

Table of Contents:

1. Project Requirements.....	1
2. Project Design.....	5
3. Project Implementation.....	19

Project Requirements:

Functional Requirements:

This application provides functionality for the user: Customer and Seller. Users will be able to create an account and submit/return various information into the Warehouse Management System. Users may edit their own account, including username and password. The UI will be designed for these users in mind, as well as to create a comprehensive view of the entire warehouse stock. As a warehouse management system, the user will be able to query, add, edit, and delete product listings inside the warehouse. If no actions are made, the system will display a list of all products. Inside these product listings, users can directly view information related to the seller and customer. This information includes seller name, seller location, customer name and address, amount paid, tracking information, etc. Users may also generate bills by selecting the product.

Functions:

Index page:

- Index page has the functionalities such as navigating to User Profile, User List, Products, Add Product, Stock, Order Tracking, and Generate Bill.

Create User:

- Users have two options for creating an account: Customer account/Seller account.
- System provides a form for users to create their account which requires email, password, Name, Address, Phone Number, and Website.

Edit User:

- Users will be able to edit their own user profile which includes verification of password, Change email, Change password, Change Name, Change Address, Change Phone Number, and Change Website.
- System will only allow editing for users if they are logged in and exist.
- After the editing is successful, the page also prints the message that user information is updated successfully.

View Profile:

- Users will be able to view profiles which display the information entered on the create user page. It also includes if the user is a customer or a seller.

Login Page:

- Users enter the username which is the email, and password to login to their account and as they login, it prints the message "Welcome "username"".

User List:

- Users select the type by customer/seller and it returns the user list which is in the database by type.

User Search Page:

- Users can search for user information by name.

Return tracking information:

- Users will be able to request the information by entering order id.
- Users will be returned tracking information of products in the warehouse system.
- The system will query and display the correct tracking information from the product the user specified along with the order and billing information.

Return customer information:

- Users will be able to view customer information of purchased products in the warehouse system.
- The system will open a modal alongside seller information, and display the customer who purchased the product, providing information such as name, address, date purchased, amount paid, etc.

Return seller information:

- Users will be able to request and be returned seller information of each product in the warehouse system.
- The system will deliver this information in a clear manner, (most likely in a modal), and displays specific and detailed information about the seller of a product, including name, location, link to company site, etc.

Generate bill:

- Users will be able to generate a bill by filling out a form with products and quantities. This can be edited or deleted before it is submitted and added to the database.
- System will update the product listings and quantities after.

Product information (Search, Sort):

- Users can select an entry to view information about the product. Each product will have an individual page displaying all information about it, such as quantity, product description, or price per unit.

- Users can search the database to find a specific product entry based on name, category, or other parameters. The system returns all the listings that match the search parameters.

Add, Update, Delete listing from database:

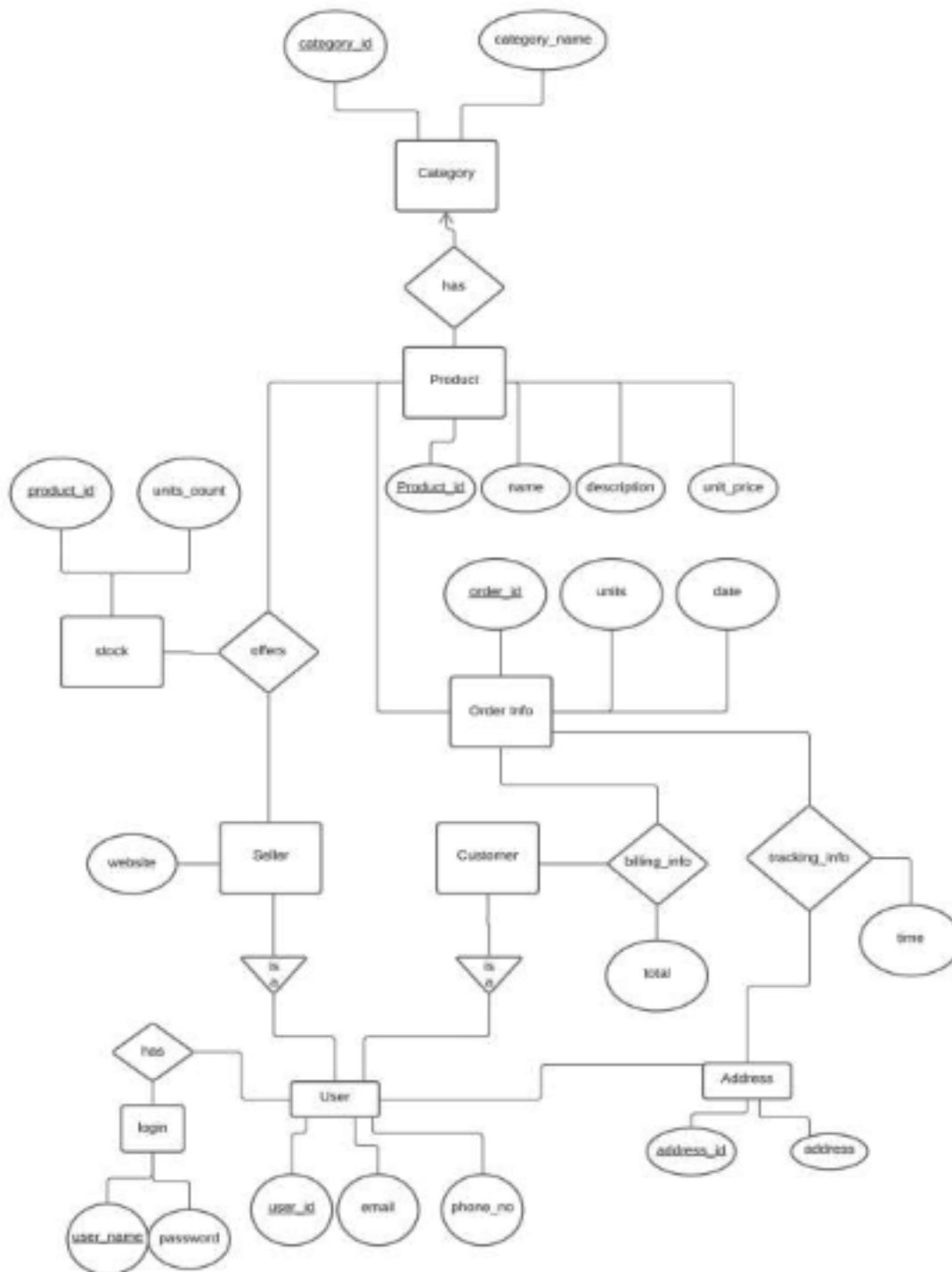
- Users will be able to add, update, and edit listings in the warehouse management system.
- System will ensure all the above requirements are met.

Return Stock information:

- Users can also see the stock information to see the quantity of the products available in the warehouse.

Project Design:

Entity-Relationship Diagram:



E/R diagram explanation:

As you can see in the diagram above, this system consists of entities: Seller, Customer, Category, User, Login, Address, Order Info, and Product.

Customer and Seller are the subclasses of User, which inherits the primary key "user_id" and attributes from the user such as "email" and "phone_no". The User has the ability to login to the system using the primary key "user_name" and "password".

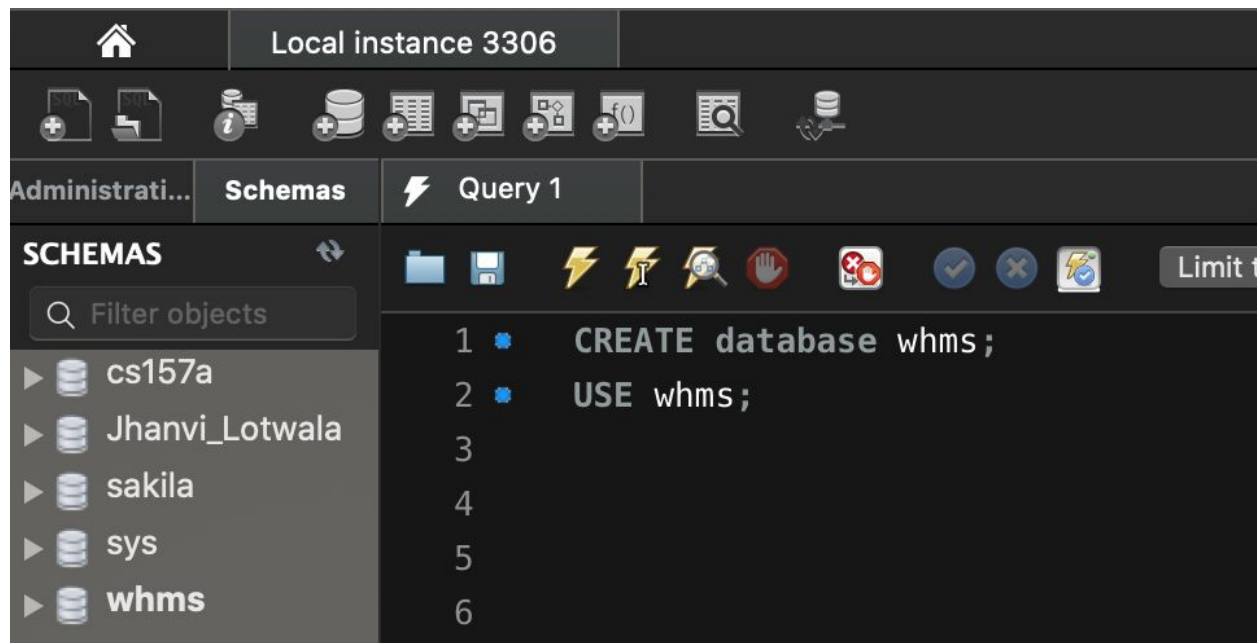
The Seller has its own attribute "website" which leads to the Seller's own company website. The Seller also "offers" a product/stock. On the other end, the Customer can request billing and tracking information. Address entity has the primary key "address_id" and "address" as the attributes and it is connected to the Order_info entity via tracking_info as a relation. Order_info has "order_id" as a primary key and "product_id", "units" ordered, and "date" as attributes.

The Product entity is another entity, with attributes "name", "description", "unit_price", and primary key "product_id". Products have "categories" which have attributes "category_id" and "category_name". These product categories can include electronics, toys, and more. The stock entity contains the primary key "product_id" and the "unit count" of that product.

Schemas:

- 1) Category(category_id, category_name)
- 2) Product(product_id, category_id, product_name, description, unit_price)
- 3) Tracking_info(order_id, date, time, address_id, address)
- 4) Order_info(order_id, user_id, product_id, category_id, units, date)
- 5) Stocks(product_id, units_count)
- 6) Offers(product_id, user_id)
- 7) Billing_info(user_id, order_id, total)
- 8) Address(address_id, address)
- 9) User(user_id, user_name, user_type, address_id, email, phone_no)
 - a) Customer(user_id, user_name)
 - b) Seller(user_id, user_name, website)
- 10) Login(user_id, email, password)

Tables with screenshots from MySql Workbench:



Creating our warehouse management database:

```
CREATE database whms;  
USE whms;
```

Category Table:

The screenshot shows the MySQL Workbench interface. On the left, the 'SCHEMAS' pane displays a tree view of databases. The 'whms' database is selected, and its 'Tables' folder is expanded, showing the 'category' table. The 'category' table's structure is visible, including columns 'category_id' and 'category_name'. The 'product' table is also visible, showing columns 'product_id', 'category_id', 'product_name', 'description', and 'unit_price'.

In the center, the 'SQL File 4*' tab is active, showing the following SQL queries:

```
63 INSERT INTO product values (1, 15
64 INSERT INTO product values (2, 15
65 INSERT INTO product values (3, 15
66
67 SELECT * from category;
```

On the right, the 'Result Grid' displays the data from the 'category' table. The table has two columns: 'category_id' and 'category_name'. The data is as follows:

category_id	category_name
1	Electronics
2	Grocery
3	Beauty
4	Health
5	Toys
6	Sports
7	Outdoors
8	Automotive
9	Pet
10	Home
11	Clothing
12	Shoes
13	Jewelry
14	Books
15	Music
NULL	NULL

Product Table:

The screenshot shows the MySQL Workbench interface. On the left, the 'SCHEMAS' pane displays a tree view of databases including 'cs157a', 'katie_nguyen', 'sakila', 'scopus_patronage', 'scopus_social_capital', 'sys', 'test', and 'whms'. Under 'whms', the 'Tables' folder is expanded, showing 'address', 'category', 'order_info', 'product', and 'tracking_info'. The 'product' table is selected. The main editor area shows the 'product - Table' with a 'Result Grid' view. The grid displays 102 rows of product data, including product_id, category_id, product_name, description, and unit_price. The data is filtered to show 50 rows.

product_id	category_id	product_name	description	unit_price
11	1	Home Camera	1080p full HD live-streaming surveillance camera	25.50
12	1	Telescope	400mm focal length and 70mm aperture, optics ...	99.99
13	1	Wireless Headset	Wireless over ear headphones with microphone...	53.50
14	1	Wireless Controller	Battery powered wireless controller, wide wirel...	49.88
15	1	Wired Keyboard	RGB backlit keyboard with ergonomic design for ...	39.99
21	2	Crackers	One 11oz box with 8 stacks of flaky crackers	1.96
22	2	Angel Hair Pasta	16oz box of angel hair pasta. Cooks in 8 minutes.	0.97
23	2	Oil	1 gallon of pure canola oil.	3.69
24	2	Medium Roast Ground Coffee	38.4oz medium roast pre-ground coffee	5.26
25	2	Tortilla Chips	9.75oz bag of crisp tortilla chips	1.98
31	3	Hand Cream	3oz tube of concentrated hand cream	3.49
32	3	Body Wash	34oz cucumber and tea scented body wash wit...	16.88
33	3	Balm	4 tubes of 0.15oz beeswax based lip balm	9.57
41	4	Zinc	100 count 50mg zinc caplets	5.00
42	4	Balm	Calendula and chamomile based balm to heal cu...	9.99
43	4	Thermometer	Touchless digital thermometer with 0.1 degree f...	22.99
51	5	Glow Sticks	100 pack multicolor waterproof glowsticks. Stay...	11.95
52	5	Tiles	32 piece clear color magnetic building tiles	49.95
53	5	Stack Cups	8 brightly colored cups stack together with num...	3.99
61	6	Loop Bands	5 resistance bands made with natural latex to in...	7.98
62	6	Brace	2 pack compression sleeve for managing knee pain	12.00
63	6	Mat	High density yoga mat with carrying strap	20.37
71	7	Headlamp	2 pack LED headlamp, AAA battery powered	13.99
72	7	Bike Light	USB rechargeable front and rear LED light	13.00
73	7	Lantern	Battery powered, waterproof, LED lantern	27.99
81	8	Oil	1qt Full synthetic motor oil	5.52
82	8	Jumper Cables	16ft, 2 gauge jump start cables with vinyl coating	29.99
83	8	Wiper Blades	2 pack 18 inch water repellency wiper blades	13.16
91	9	Trash Bags	8 rolls of 15 count 9 by 13 inch doggy waste bags	6.99
92	9	Cat Litter	18lb hard clumping, dust free cat litter	9.99
93	9	Dry Dog Food	3.8 dry dog food for small adult dogs	7.49
101	10	Stack Cups	20 pack recyclable plastic 16oz stackable cups	4.99
102	10	Trash Bags	10 rolls of 15 count 13 gallon trash bags	10.99

Tracking_info table:

The screenshot shows a database management interface with a dark theme. The top bar indicates 'Local instance 3306'. The left sidebar shows a tree view of schemas, with 'whms' expanded to show 'Tables' and 'Tracking_info'. The main editor displays SQL code for creating the 'Tracking_info' table and inserting data. The 'Result Grid' at the bottom shows 15 rows of data.

SQL Code:

```
1 CREATE TABLE whms.Tracking_info (
2     order_id INT NOT NULL,
3     date DATE NOT NULL,
4     time TIME NOT NULL,
5     address_id INT NOT NULL,
6     address VARCHAR(80) NOT NULL,
7     PRIMARY KEY (order_id));
8
9 INSERT INTO Tracking_info VALUES (1,'2020-10-09','08:05:56',1,"12,Main St,San Jose,CA-95125");
10 INSERT INTO Tracking_info VALUES (2,'2020-10-09','08:07:50',2,"2,Rose St,San Jose,CA-95125");
11 INSERT INTO Tracking_info VALUES (3,'2020-10-09','08:09:10',3,"10,Rowan St,Santa Clara,CA-95050");
12 INSERT INTO Tracking_info VALUES (4,'2020-10-09','09:05:49',4,"23,Babylon St,San Jose,CA-95123");
13 INSERT INTO Tracking_info VALUES (5,'2020-10-09','09:15:31',5,"21,Newbee St,San Jose,CA-95123");
14 INSERT INTO Tracking_info VALUES (6,'2020-10-09','09:20:56',6,"5,Roseline St,San Jose,CA-95123");
15 INSERT INTO Tracking_info VALUES (7,'2020-10-09','10:05:40',7,"10,Adobe St,San Jose,CA-95134");
16 INSERT INTO Tracking_info VALUES (8,'2020-10-09','10:07:34',8,"9,Main St,San Jose,CA-95125");
17 INSERT INTO Tracking_info VALUES (9,'2020-10-09','10:09:56',9,"4,Babylon St,San Jose,CA-95123");
18 INSERT INTO Tracking_info VALUES (10,'2020-10-09','10:10:23',10,"14,Rowan St,Santa Clara,CA-95050");
19 INSERT INTO Tracking_info VALUES (11,'2020-10-09','10:11:35',11,"12,Newbee St,San Jose,CA-95123");
20 INSERT INTO Tracking_info VALUES (12,'2020-10-09','10:13:50',12,"21,Rose St,San Jose,CA-95125");
21 INSERT INTO Tracking_info VALUES (13,'2020-10-09','10:14:56',13,"76,Main St,San Jose,CA-95125");
22 INSERT INTO Tracking_info VALUES (14,'2020-10-09','10:30:50',14,"75,Main St,San Jose,CA-95125");
23 INSERT INTO Tracking_info VALUES (15,'2020-10-09','10:32:32',15,"13,Brian St,Santa Clara,CA-95050");
```

Result Grid:

order_id	date	time	address_id	address
1	2020-10-09	08:05:56	1	12,Main St,San Jose,CA-95125
2	2020-10-09	08:07:50	2	2,Rose St,San Jose,CA-95125
3	2020-10-09	08:09:10	3	10,Rowan St,Santa Clara,CA-95050
4	2020-10-09	09:05:49	4	23,Babylon St,San Jose,CA-95123
5	2020-10-09	09:15:31	5	21,Newbee St,San Jose,CA-95123
6	2020-10-09	09:20:56	6	5,Roseline St,San Jose,CA-95123
7	2020-10-09	10:05:40	7	10,Adobe St,San Jose,CA-95134
8	2020-10-09	10:07:34	8	9,Main St,San Jose,CA-95125
9	2020-10-09	10:09:56	9	4,Babylon St,San Jose,CA-95123
10	2020-10-09	10:10:23	10	14,Rowan St,Santa Clara,CA-95050
11	2020-10-09	10:11:35	11	12,Newbee St,San Jose,CA-95123
12	2020-10-09	10:13:50	12	21,Rose St,San Jose,CA-95125
13	2020-10-09	10:14:56	13	76,Main St,San Jose,CA-95125
14	2020-10-09	10:30:50	14	75,Main St,San Jose,CA-95125
15	2020-10-09	10:32:32	15	13,Brian St,Santa Clara,CA-95050

Order_info table:

Administrati... Schemas Query 1 Order_info offers Billing_info Tracking_info

SCHEMAS

Filter objects

cs157a Jhanvi_Lotwala sakila sys whms

Tables

Billing_in... offers Order_info Tracking...

Views Stored Pro... Functions

Object Info Session

No object selected

```

3 CREATE TABLE whms.Order_info(
4     order_id INT NOT NULL,
5     user_id INT NOT NULL,
6     product_id INT NOT NULL,
7     category_id INT NOT NULL,
8     units INT NOT NULL,
9     date DATE NOT NULL,
10    PRIMARY KEY (order_id));
11
12 INSERT INTO Order_info VALUES(1,1,11,3,2,'2020-10-09');

```

100% 57:25

Result Grid Filter Rows: Search Edit: Export/Import:

order_id	user_id	product_id	category_id	units	date
1	1	11	3	2	2020-10-09
2	1	12	4	3	2020-10-09
3	1	13	3	2	2020-10-09
4	2	14	3	200	2020-10-09
5	2	15	3	300	2020-10-09
6	3	21	4	500	2020-10-09
7	3	22	1	400	2020-10-09
8	4	23	6	200	2020-10-09
9	5	24	6	300	2020-10-09
10	6	25	7	6	2020-10-10

Stocks Table:

The screenshot shows a database management interface with a dark theme. The left sidebar displays a tree view of schemas, including 'cs157a', 'Jhanvi_Lotwala', 'sakila', 'sys', and 'whms'. Under 'whms', the 'Tables' folder is expanded, showing 'Billing_in...', 'Order_info', 'stocks', and 'Tracking...'. The main editor area displays SQL code for creating the 'stocks' table and inserting data. The code is as follows:

```

3 CREATE TABLE whms.stocks(
4     product_id INT NOT NULL,
5     units_count INT NOT NULL);
6
7 INSERT INTO stocks VALUES(11,50000);
8 INSERT INTO stocks VALUES(12,30000);
9 INSERT INTO stocks VALUES(13,20000);
10 INSERT INTO stocks VALUES(14,20000);
11 INSERT INTO stocks VALUES(15,30000);
12 INSERT INTO stocks VALUES(21,50000);
13 INSERT INTO stocks VALUES(22,40000);
  
```

Below the SQL editor, the 'Result Grid' is visible, showing the data inserted into the 'stocks' table. The grid has two columns: 'product_id' and 'units_count'. The data is as follows:

product_id	units_count
11	50000
12	30000
13	20000
14	20000
15	30000
21	50000
22	40000
23	20000
24	30000
25	60000
31	40000
32	50000
33	40000

The bottom of the interface shows the 'Object Info' tab, which is currently empty, displaying 'No object selected'.

Address table:

Filter objects

- cs157a
- katie_nguyen
- sakila
- scopus_patronage
- scopus_social_capital
- sys
- test
- whms
 - Tables
 - address
 - category
 - login
 - product
 - Views
 - Stored Procedures
 - Functions
- world

Administration Schemas

Information

No object selected

```

323 • INSERT INTO address VALUES(3, "10,Rowan St,Santa Clara,CA-95050");
324 • INSERT INTO address VALUES(4, "23,Babylon St,San Jose,CA-95123");
325 • INSERT INTO address VALUES(5, "21,Newbee St,San Jose,CA-95123");
326 • INSERT INTO address VALUES(6, "5,Roseline St,San Jose,CA-95123");
327 • INSERT INTO address VALUES(7, "10,Adobe St,San Jose,CA-95134");
328 • INSERT INTO address VALUES(8, "9,Main St,San Jose,CA-95125");
329 • INSERT INTO address VALUES(9, "4,Babylon St,San Jose,CA-95123");

```

Result Grid

address_id	address
1	12 Main St,San Jose,CA-95125
2	2,Rose St,San Jose,CA-95125
3	10,Rowan St,Santa Clara,CA-95050
4	23,Babylon St,San Jose,CA-95123
5	21,Newbee St,San Jose,CA-95123
6	5,Roseline St,San Jose,CA-95123
7	10,Adobe St,San Jose,CA-95134
8	9,Main St,San Jose,CA-95125
9	4,Babylon St,San Jose,CA-95123
10	14,Rowan St,Santa Clara,CA-95050
11	12,Newbee St,San Jose,CA-95123
12	21,Rose St,San Jose,CA-95125
13	76,Main St,San Jose,CA-95125
14	75,Main St,San Jose,CA-95125
15	13,Brian St,Santa Clara,CA-95050
16	111 10th St, San Jose, CA-95192
17	50 24th St, San Jose, CA-95192
18	300 Aborn Rd, San Jose, CA-95121
19	88 S King Rd, San Jose, CA-95121
20	100 Blossom Hill Rd, San Jose, CA-...
21	888 Tully Rd, San Jose, CA-95121
*	NULL

User Table:

MySQL Workbench

cs157a x

File Edit View Query Database Server Tools Scripting Help

Navigator

SQL File 3* HW2 157proj x

Limit to 50 rows

115 INSERT INTO user VALUES (29, 'Thomas Shirley', 'seller', 21, 'thomasshirley@email.o

116 INSERT INTO user VALUES (30, 'Tob Sibley', 'seller', 21, 'tobsibley@email.com', 916

117 INSERT INTO user VALUES (31, 'Steven Silver', 'seller', 21, 'stevens@email.org', 91

118 INSERT INTO user VALUES (32, 'Ju Myung Song', 'seller', 17, 'songjm@email.com', 916

119

120 • SELECT * FROM user;

Result Grid

Filter Rows:

Edit: Export/Import: Wrap Cell Content:

	user_id	user_name	user_type	address_id	email	phone_no
1	Mike Wu	customer	1	mikewu@email.com	4081234567	
2	John Smith	customer	2	johnsmith@email.com	4082222222	
3	Debbie Abbott	customer	3	abbotdebbie@email.com	4083333333	
4	Vivek Agarwal	customer	4	vivekagarwal@email.com	4084084444	
5	Nitin Aggarwal	customer	4	nittinagarwal@email.com	4089999988	
6	Leslie Albert	customer	5	lesliealbert@email.com	4089753455	
7	Thomas Altura	customer	6	thomasaltura@email.com	5103454353	
8	Usha Andra	customer	7	ushaandra@email.com	5105858585	
9	Michael Ashley	customer	8	michaelashley@email.com	5105105100	
10	Anu Basu	customer	9	anubasu@email.com	5109412344	
11	Michael Beasley	customer	10	michaelb@email.com	5100987678	
12	Roy Blitzler	customer	11	royb@email.com	5102938475	
13	Mi Nan Boyd	customer	12	minanb@email.com	91693948527	
14	Helen Brubeck	customer	12	hbrubeck@email.com	9163872435	
15	Joel Busch	customer	13	joelbusch@email.com	9169161111	
16	Nancy Bussani	seller	14	nbussani@email.org	9164433246	
17	Mary Calegari	seller	15	maryc@email.com	9165837567	
18	BJ Campsey	seller	16	bjcampsey@email.com	9162314234	
19	Caroline Chen	seller	17	carolchen@email.com	9161234567	
20	Thomas Chandy	seller	18	thomaschandy@email.com	4081231231	
21	Yu Chen	seller	18	yuchen@email.com	4081231232	
22	Jang Cho	seller	18	jangcho@email.com	4081231233	
23	Singmay Chou	seller	19	singmayc@email.com	4087654321	
24	Howard Combs	seller	19	hcombs@email.com	4089999944	
25	Sharawn Con...	seller	19	sconners@email.com	4083933533	
26	Laura Roden	seller	20	lauraroden@email.org	9161112223	
27	Ron Roman	seller	21	ronroman@email.org	9162948385	
28	Jean Sawyer	seller	21	jsawyer@email.org	9168932784	
29	Thomas Shirley	seller	21	thomasshirley@email.org	9168277777	
30	Tob Sibley	seller	21	tobsibley@email.com	9168888888	
31	Steven Silver	seller	21	stevens@email.org	9169999999	
32	Ju Myung Song	seller	17	songjm@email.com	9169299999	

user 1 x

Output

Action Output

Administration Schemas

Information

Table: user

Columns:

user_id int(11) PK

user_name varchar(45)

user_type varchar(45)





address_id varchar(45)

email varchar(45)

phone_no int(11)

Customer Table:

```
120 • INSERT INTO customer(user_id, user_name)
121     SELECT user_id, user_name
122     FROM user
123     WHERE user_type like 'customer';
124
125 • select * from customer;
```






<   Filter Rows: Edit:  

	user_id	user_name
▶	1	Mike Wu
	2	John Smith
	3	Debbie Abbott
	4	Vivek Agarwal
	5	Nitin Aggarwal
	6	Leslie Albert
	7	Thomas Altura
	8	Usha Andra
	9	Michael Ashley
	10	Anu Basu
	11	Michael Beasley
	12	Roy Blitzer
	13	Mi Nan Boyd
	14	Helen Brubeck
	15	Joel Busch
•	NULL	NULL

Seller Table:

```
161
162 • select * from seller;
```

<

Result Grid   Filter Rows: Edit:    Export/Imp

	user_id	user_name	website
▶	16	Nancy Bussani	dtsjgoods.com
	17	Mary Calegari	sanjosesport.com
	18	BJ Campsey	homegoods.com
	19	Caroline Chen	petexpress.com
	20	Thomas Chandy	audiosj.com
	21	Yu Chen	audiosj.com
	22	Jang Cho	audiosj.com
	23	Singmay Chou	dtsjsupply.com
	24	Howard Combs	dtsjsupply.com
	25	Sharawn Connors	dtsjsupply.com
	26	Laura Roden	repairsupplies.com
	27	Ron Roman	baygrocery.com
	28	Jean Sawyer	baygrocery.com
	29	Thomas Shirley	baygrocery.com
	30	Tob Sibley	baygrocery.com
	31	Steven Silver	baygrocery.com
	32	Ju Myung Song	books.org
•	NULL	NULL	NULL

Login Table:

199
200 ● `SELECT * FROM login;`
201

Result Grid Filter Rows: Edit: Export/Imp

	user_id	email	password
▶	1	mikewu@email.com	admin
	2	johnsmith@email.com	123456
	3	abbotdebbie@email.com	qwerty
	4	vivekagarwal@email.com	ldog
	5	nittinagarwal@email.com	Corn
	6	lesliealbert@email.com	pho
	7	thomasaltura@email.com	2020class
	8	ushaandra@email.com	Destinys_Child
	9	michaelashley@email.com	pass
	10	anubasu@email.com	spotify
	11	michaelb@email.com	aaaaaaaaaa
	12	royb@email.com	andre
	13	minanb@email.com	3000qqqqqq
	14	hbrubeck@email.com	hgfdigtkre
	15	joelbusch@email.com	seller
	16	nbussani@email.org	underLine
	17	maryc@email.com	computeR
	18	bjcampsey@email.com	neopets
	19	carolchen@email.com	Estelise
	20	thomaschandy@email.com	phOne
	21	yuchen@email.com	sjsu20
	22	jangcho@email.com	Chere
	23	singmayc@email.com	banh
	24	hcombs@email.com	30248765487...
	25	scanners@email.com	line

login 6 x

:(11) PK
rchar(45)

Billing Info Table:

The screenshot shows a database management interface with the following components:

- Top Bar:** Home icon, database name 'CS157ASJSU', and various tool icons.
- Navigation Tabs:** 'Administration' and 'Schemas' (selected).
- Left Panel (SCHEMAS):** A tree view showing the database structure. The 'whms' schema is expanded, showing 'Tables' (including 'billing_info'), 'Views', 'Stored Procedures', and 'Functions'.
- Center Panel:** A query editor showing the query: `SELECT * from billing_info`.
- Bottom Panel:**
 - Object Info:** Shows 'Schema: whms'.
 - Session:** Empty.
 - Result Grid:** Displays the query results in a table format.

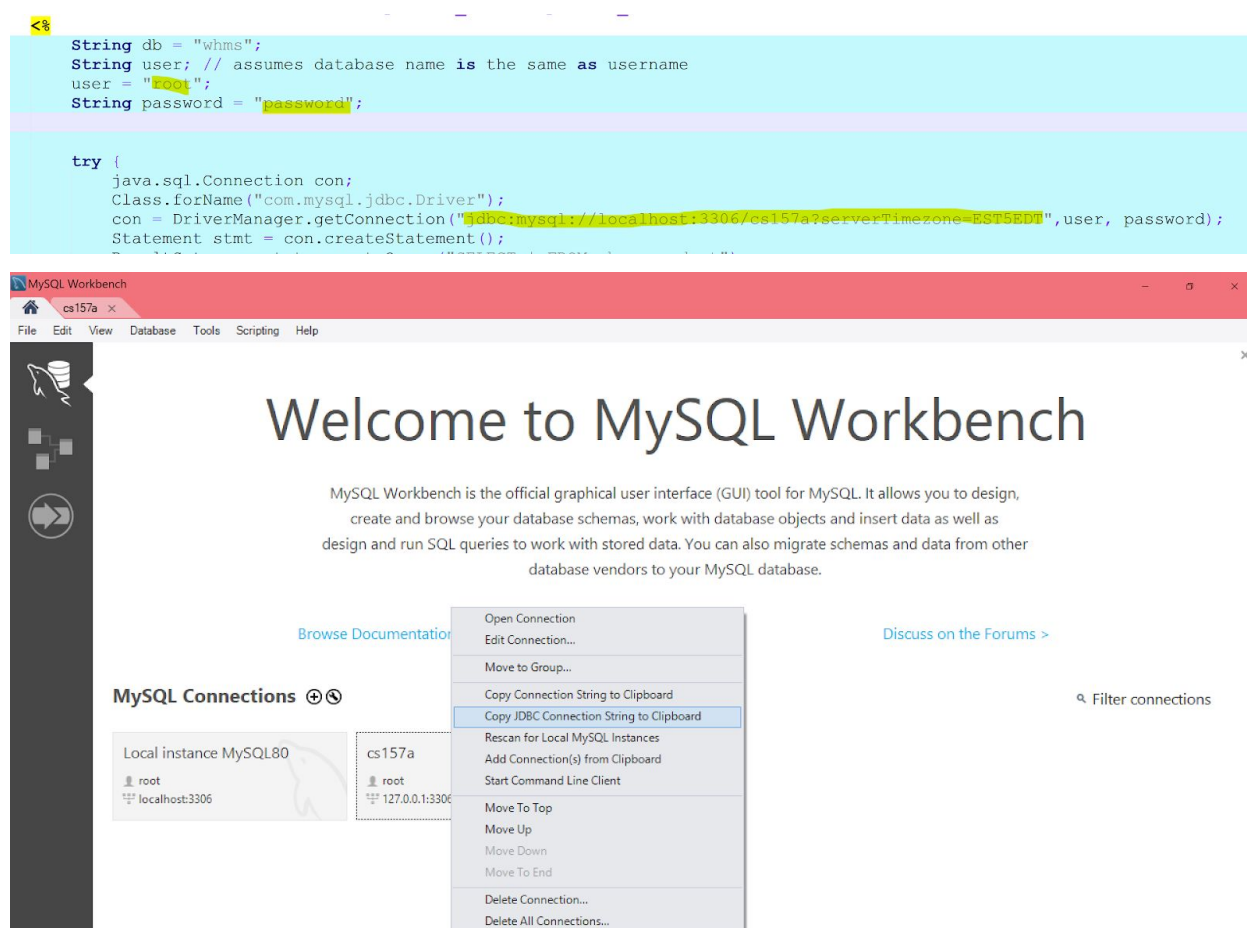
Result Grid Data:

	order_id	user_id	total
▶ 1	1	300	
2	2	500	
3	1	500	
4	5	1000	
5	6	50	
6	2	120	
7	3	150	
8	4	2250	
9	22	250	
10	2	2350	
11	1	350	
12	3	12	
13	1	5	
14	5	522	
15	1	2222	
	NULL	NULL	NULL

Project Implementation:

We started with html and css to set the layout of our warehouse management system. We have the following web pages: index.html, login.html, register.html, editprofile.html, orders.html, products.html, search.html, and users.html. We used styles.css for fonts and color. We used jsp to make these web pages functional. These are the jsp files: login.jsp, editprofile.jsp, orders.jsp, products.jsp, register.jsp, search.jsp, addProduct.jsp, addProductComplete.jsp, recent_orders.jsp, trackOrder.jsp, update.jsp, updateComplete.jsp, userpage.jsp, users.jsp, generateBillComplete.jsp, and generateBill.jsp. We used JDBC to connect our MySQL database to the ApacheTomcat server.

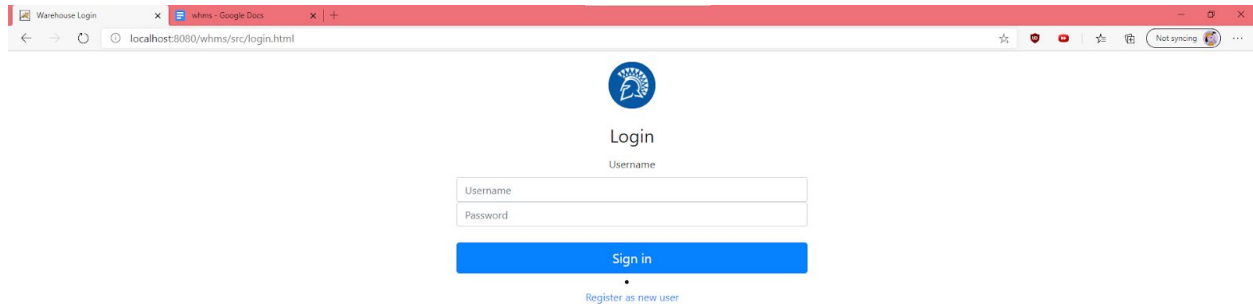
To set up the system, a new database must be set up in MySQL. The tables for the database can be added by running the attached “whms db” SQL script. For all of the JSP files, the user must update the user and password for their own database. The user may need to update the JDBC connection string as well. This can be found from the homepage of MySQL Workbench.



Once the database is set up, the user may run the system using Apache Tomcat.

Users must be logged in to access features. The first page that users will see is the login page. Users have the option to register if they do not already have an account.

Login page:



Warehouse Login

whms - Google Docs

localhost:8080/whms/src/login.html

Warehouse

Login

Username

Username

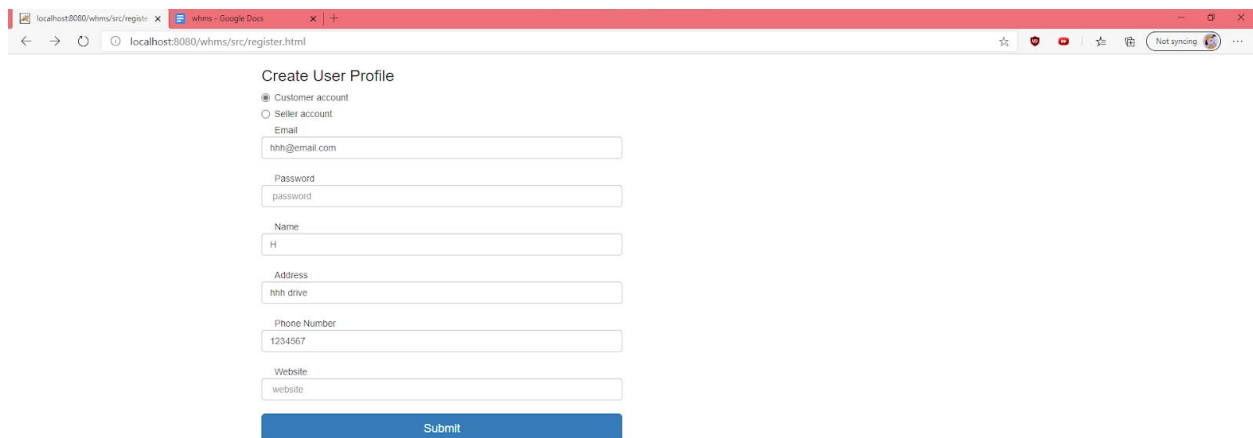
Password

Sign in

Register as new user

Create User

Users have the option to register as either a customer or a seller. Email, password, name, address, and phone number are required. If the user is registering as a seller, they may include a website.



localhost:8080/whms/src/register.html

Warehouse

Create User Profile

Customer account

Seller account

Email

hjh@email.com

Password

password

Name

H

Address

hjh drive

Phone Number

1234567

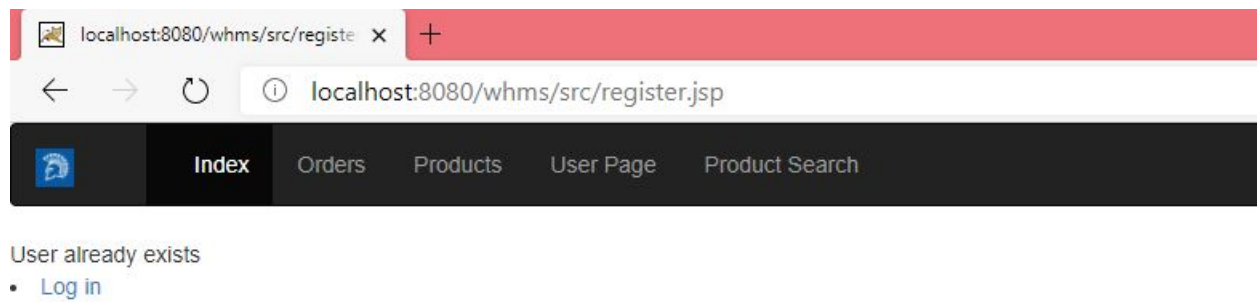
Website

website

Submit

The user inputted data is forwarded to register.jsp. Register.jsp makes a connection to the mysql server and checks whether or not a user with the inputted email exists. If the email entered is already registered, then users will not be able to complete the

registration process.



If the user registration is a success, the email will be used as the username and the user will be logged in for the current session.



The user input inserted into the login, address, and user tables in the mysql server. User_id and address_id are automatically generated.

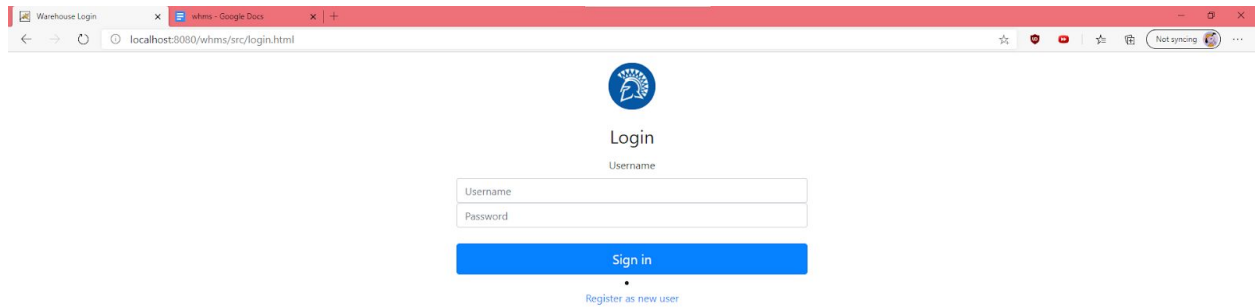
user_id	email	password
42	l	l
43	g	g
44	q	q
45	r	w
46	hhhhh	h
47	j	j
48	p@email.com	p
49	Mark@mark.com	mark
50	email	e
51	newemail	a
52	njdkfwlt	p
53	jhanvi	fffff
54	hhh@email.com	h
NULL	NULL	NULL

address_id	address
22	a
23	g
24	q
25	w
26	h
27	j
28	p
29	111 Street st.
30	1234 street
31	address
32	new address
33	sg
34	hhh drive
NULL	NULL

user_id	user_name	user_type	address_id	email	phone_no	website
32	Ju Myung Song	seller	17	songjm@email.com	9169299999	petexp...
43	g	customer	23	g	g	NULL
44	q	seller	24	q	q	NULL
45	w	seller	25	r	w	NULL
46	h	seller	26	hhhhh	h	NULL
47	Joseph	customer	27	j	1234	NULL
48	pppp	customer	28	p@email.com	111	NULL
49	Mark	customer	29	Mark@mark.com	4081111111	NULL
50	john	customer	30	email	22	NULL
51	name	customer	31	newemail	123	NULL
52	name	customer	32	njdkfwlt	1234	NULL
53	jhanvi	customer	33	jhanvi	1111	NULL
54	H	customer	34	hhh@email.com	1234567	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL

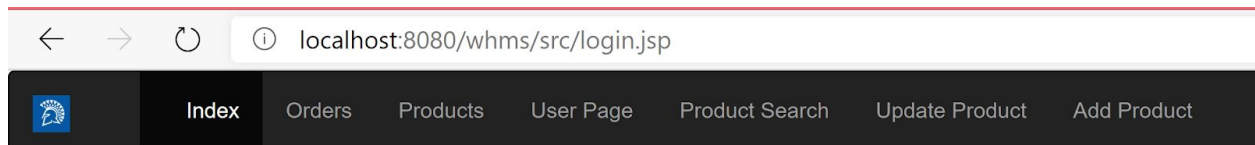
As you can see in the database, the new user is registered when we created the user profile.

If the user already has an existing account, they may log in from the login page. Once the user is logged in, they may view their orders and user profile from the user page.



A screenshot of a web browser showing the 'Warehouse Login' page. The browser's address bar displays 'localhost:8080/whms/src/login.html'. The page features a blue circular logo with a white bird-like icon at the top center. Below the logo, the word 'Login' is centered. Underneath 'Login', the word 'Username' is displayed above a text input field. Below the 'Username' field is a 'Password' field. A blue button labeled 'Sign in' is positioned below the password field. At the bottom of the page, there is a link that reads 'Register as new user'.

Login success landing page



A screenshot of the 'Warehouse Login' success landing page. The browser's address bar shows 'localhost:8080/whms/src/login.jsp'. The page has a dark navigation bar at the top with a blue circular logo on the left and several menu items: 'Index', 'Orders', 'Products', 'User Page', 'Product Search', 'Update Product', and 'Add Product'. Below the navigation bar, the text 'Welcome lesliealbert@email.com' is displayed.

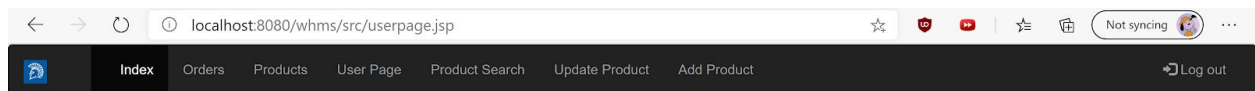
Welcome lesliealbert@email.com

Return User Information:

When the user accesses the user page, userpage.jsp will execute sql queries to get the user profile information and recent order information from the user, address, and order_info tables.



User page



User Profile

Email: lesicalbert@email.com
 Name: Bryan
 Phone number: 1111
 User type: customer
 Address: 8888 7th Street, San Jose, CA 95121
 • [Edit Profile](#)

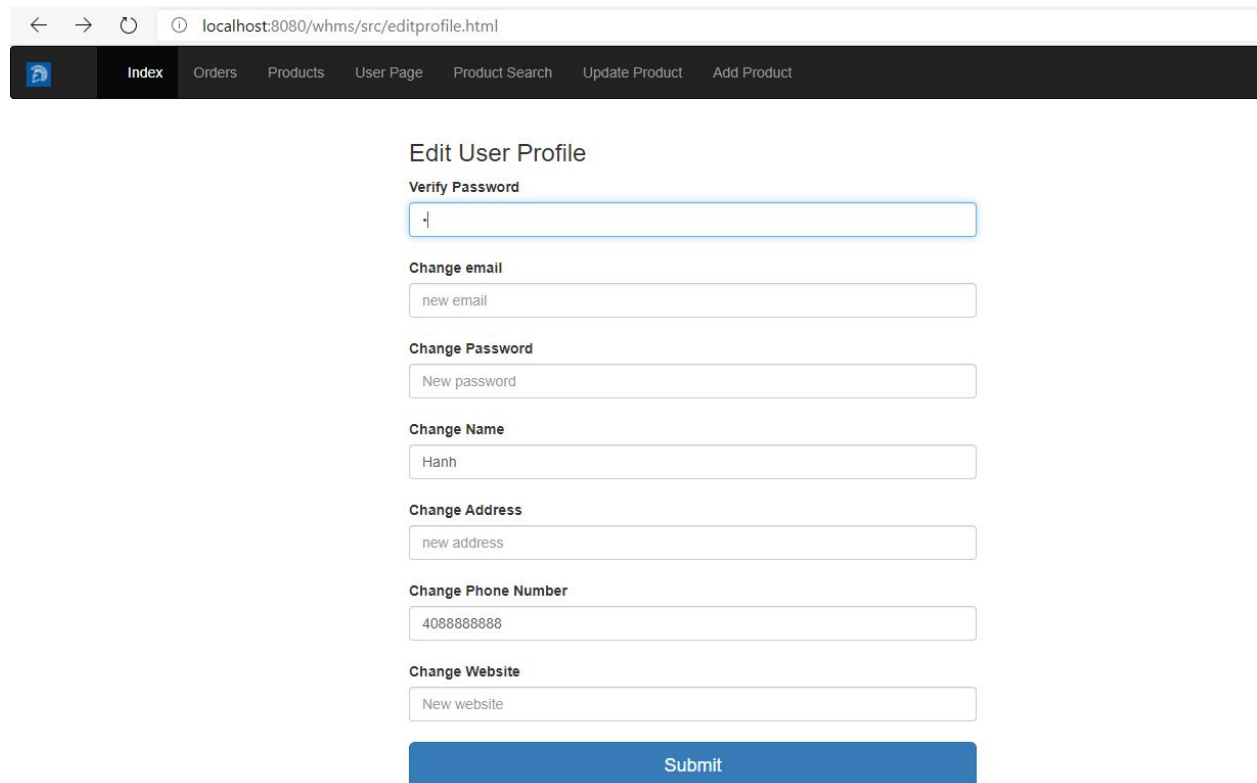
Recent orders

Order #	Product ID	Category ID	Units	Date
10	25	7	6	2020-10-09
28	11	7	2	2020-12-02
29	11	7	3	2020-12-02
30	141	14	5	2020-12-02
31	160	1	6	2020-12-02
32	21	2	10	2020-12-02
33	161	6	5	2020-12-02
34	31	3	20	2020-12-02
35	162	1	5	2020-12-02
37	122	12	8	2020-12-04
43	11	1	1	2020-12-10

Edit User:

Users can edit their own user profile, including their username, password, address, phone number, and name. The user must verify their password, otherwise the updates will not be made to the database. The user input from editprofile.html is received by editprofile.jsp, which will execute the SQL statements.

The user that was logged in to the system in the above example will change their name and phone number.



The screenshot shows a web browser window with the address bar displaying 'localhost:8080/whms/src/editprofile.html'. The page has a dark navigation bar with links: Index, Orders, Products, User Page, Product Search, Update Product, and Add Product. The main content area is titled 'Edit User Profile' and contains several form fields:

- Verify Password:** A text input field with a blue border and a cursor.
- Change email:** A text input field with the placeholder text 'new email'.
- Change Password:** A text input field with the placeholder text 'New password'.
- Change Name:** A text input field with the placeholder text 'Hanh'.
- Change Address:** A text input field with the placeholder text 'new address'.
- Change Phone Number:** A text input field with the placeholder text '4088888888'.
- Change Website:** A text input field with the placeholder text 'New website'.

At the bottom of the form is a large blue button labeled 'Submit'.

localhost:8080/whms/src/editprofile.jsp

Index Orders Products User Page Product Search Update Product Add Product

Name Updated Successfully
Phone Number Updated Successfully

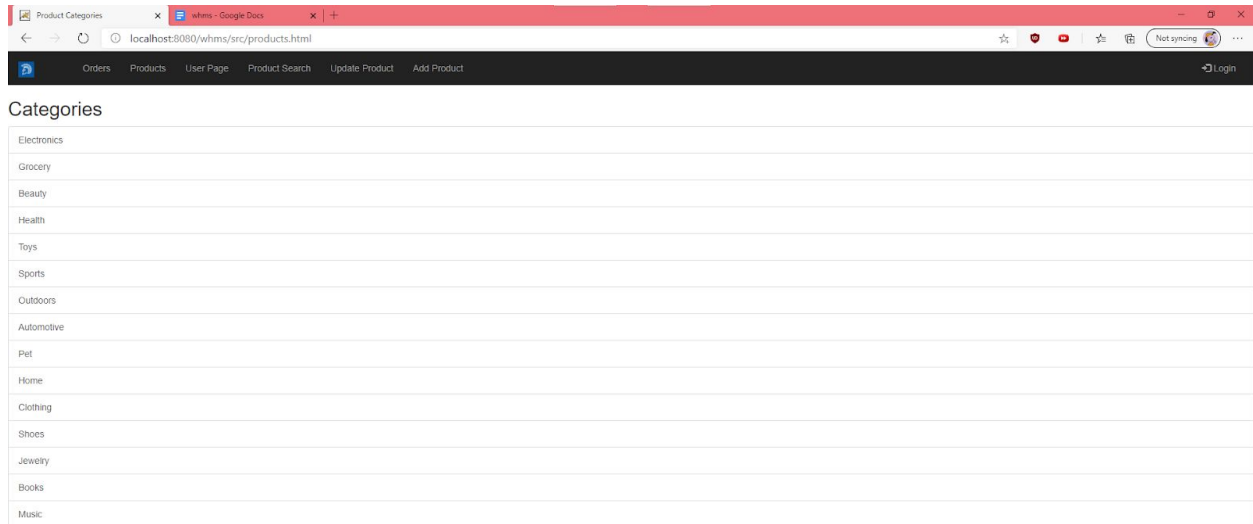
- [return to user profile](#)

	user_id	user_name	user_type	address_id	email	phone_no	website
	4	Vivek Agarwal	customer	4	vivekagarwal@email.com	4084084444	NULL
	5	Nitin Aggarwal	customer	4	nittinaggarwal@email.com	4089999988	NULL
▶	6	Hanh	customer	5	lesliealbert@email.com	4088888888	NULL
	7	Thomas Altura	customer	6	thomasaltura@email.com	5103454353	NULL
	8	Usha Andra	customer	7	ushaandra@email.com	5105858585	NULL
	9	Michael Ashley	customer	8	michaelashley@email.com	5105105100	NULL
	10	Anu Basu	customer	9	anubasu@email.com	5109412344	NULL
	11	Michael Beasley	customer	10	michaelb@email.com	5100987678	NULL

The users information has now been updated in the database.

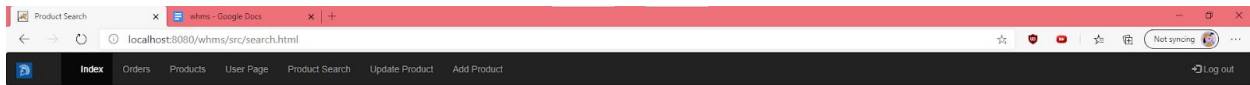
Product Information:

On the products.html page, users can select a category to view products. Selecting a category will send a request to products.jsp where a SQL query will be executed to list product information of all products in the selected category.



Products search page

From the product search page, users are able to select a search by name or description from the drop down menu. The user may sort the results by price, product id, or by name. The user input is forwarded to search.jsp, which will execute the SQL query based on the user selection.



Search Results

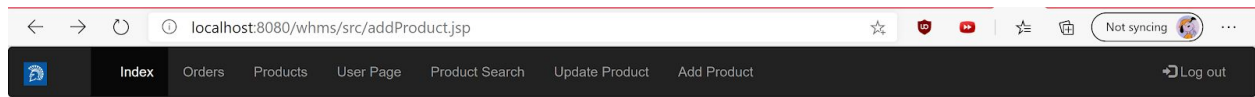
Search results for description containing: wireless

sort by: unit_price

Product ID	Category Name	Product Name	Description	Unit Price
155	Electronics	Controller	Battery powered wireless controller, wide wireless range using Bluetooth technology	49.01
113	Electronics	head	Wireless over ear headphones with microphone. Connects wirelessly with bluetooth	53.50

Add, Update, Delete Products

Users may add a new product from the add product page. The user must select a category and input a product name, description, and unit price. The product id will be automatically generated.



Add product

Beauty

Sunblock

SPF50+ waterproof sunblock

10

Submit



Adding product complete

• Add new product

	product_id	category_id	product_name	description	unit_price
	156	2	Kale	1 pound of fresh dino Kale	1.50
	157	2	Kale	Curly kale	1.20
	159	1	electronic	rdfwe	20.00
	160	1	name	description	1111.00
	161	6	Telescope	2222222222	1.02
	164	2	Beans	garbanzo beans	1.50
	165	3	Sunblock	SPF50+ waterproof sunblock	10.00
	NULL	NULL	NULL	NULL	NULL

Here, we can see that the product was added to the database. Users can check by viewing the products page or product search page.

Users can update and delete product listings from update.jsp page. Users must select the Update product or Delete product action as well as the product from the drop down menus. A SQL query is executed to retrieve the product id and name of the products for the dropdown menu. The user may select a new category from the drop down menu, input a new name, description, or unit price.



Update product

Update product

11 - Camera

Update category

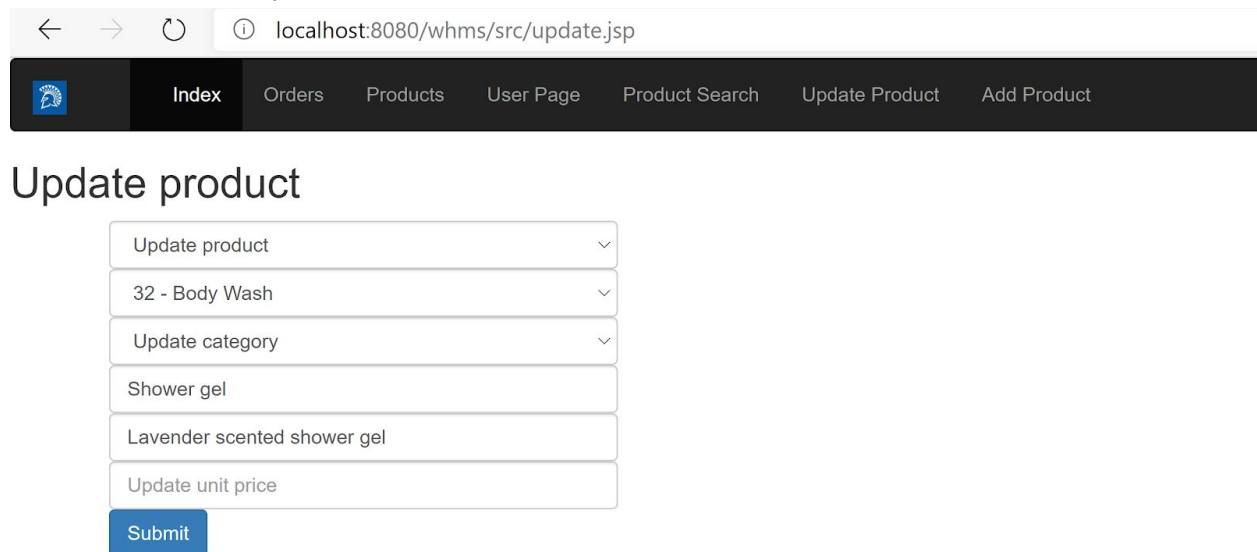
Update product name

Update description

Update unit price

Submit

In this example, Body Wash is selected.



Update product

32 - Body Wash

Update category

Shower gel


Lavender scented shower gel

Update unit price

Submit

	product_id	category_id	product_name	description	unit_price
	23	2	Oil	1 gallon of pure canola oil.	3.69
	24	2	Medium Roast Ground Coffee	38.4oz medium roast pre-ground coffee	5.26
	25	2	Tortilla Chips	9.75oz bag of crisp tortilla chips	1.98
	31	3	Hand Cream	3oz tube of concentrated hand cream	3.49
▶	32	3	Body wash	lavender liquid body wash	16.88
	33	3	Balm	4 tubes of 0.15oz beeswax based lip balm	9.57
	41	4	Zinc	100 count 50mg zinc caplets	5.00
	42	4	Balm	Calendula and chamomile based balm to heal cu...	9.99

←
→
↺
localhost:8080/whms/src/updateComplete.jsp


Index
Orders
Products
User Page
Product Search
Update Product
Add Product

Update Status

Product name Updated Successfully to: Shower gel

description Updated Successfully

- [update another product](#)
- [add a product](#)

We can see that the new information has been updated in the database.

	product_id	category_id	product_name	description	unit_price
	23	2	Oil	1 gallon of pure canola oil.	3.69
	24	2	Medium Roast Ground Coffee	38.4oz medium roast pre-ground coffee	5.26
	25	2	Tortilla Chips	9.75oz bag of crisp tortilla chips	1.98
	31	3	Hand Cream	3oz tube of concentrated hand cream	3.49
	32	3	Shower gel	Lavender scented shower gel	16.88
	33	3	Balm	4 tubes of 0.15oz beeswax based lip balm	9.57
	41	4	Zinc	100 count 50mg zinc caplets	5.00
	42	4	Balm	Calendula and chamomile based balm to heal cu...	9.99

Now returning to the same update product page, Shower gel will be deleted.

localhost:8080/whms/src/update.jsp

Index Orders Products User Page Product Search Update Product Add Product

Update product

Delete product

32 - Body wash

Update category

Update product name

Update description

Update unit price

Submit

localhost:8080/whms/src/updateComplete.jsp

Index Orders Products User Page Product Search Update Product Add Product

Update Status

Data Deleted Successfully!

- [update another product](#)
- [add a product](#)

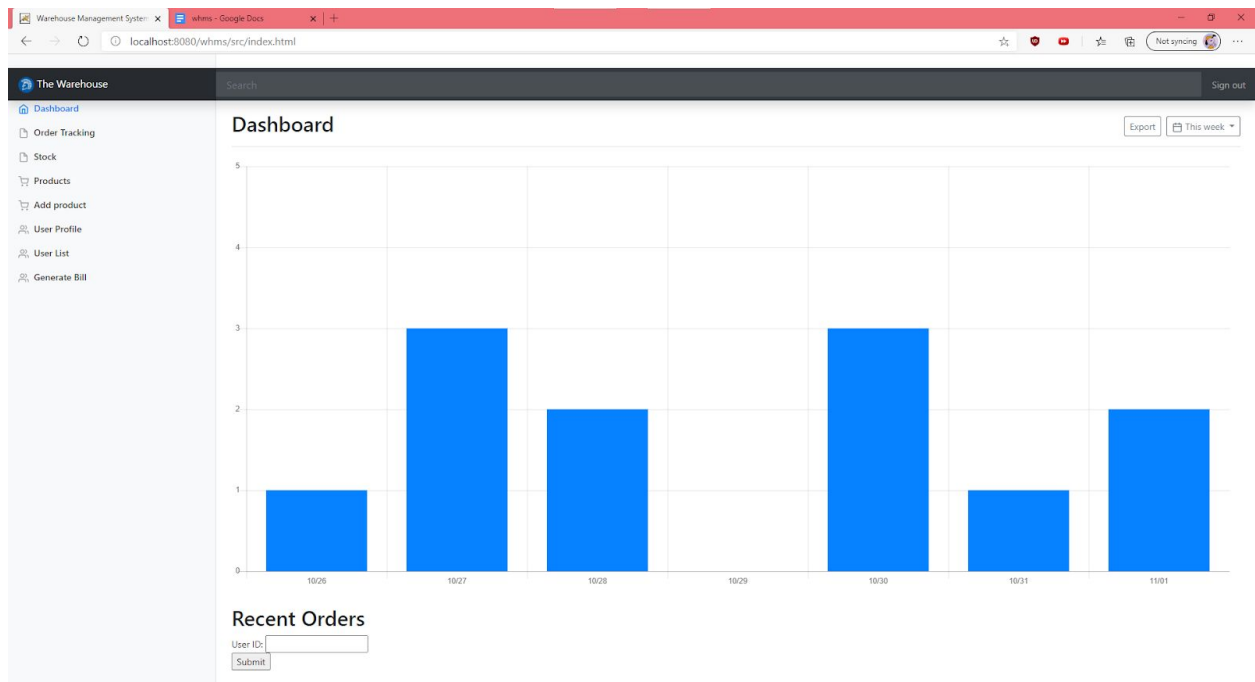
The product has been successfully deleted from the database.

	product_id	category_id	product_name	description	unit_price
	23	2	Oil	1 gallon of pure canola oil.	3.69
	24	2	Medium Roast Ground Coffee	38.4oz medium roast pre-ground coffee	5.26
	25	2	Tortilla Chips	9.75oz bag of crisp tortilla chips	1.98
	31	3	Hand Cream	3oz tube of concentrated hand cream	3.49
▶	33	3	Balm	4 tubes of 0.15oz beeswax based lip balm	9.57
	41	4	Zinc	100 count 50mg zinc caplets	5.00
	42	4	Balm	Calendula and chamomile based balm to heal cu...	9.99
	43	4	Thermometer	Touchless digital thermometer with 0.1 degree f...	22.99

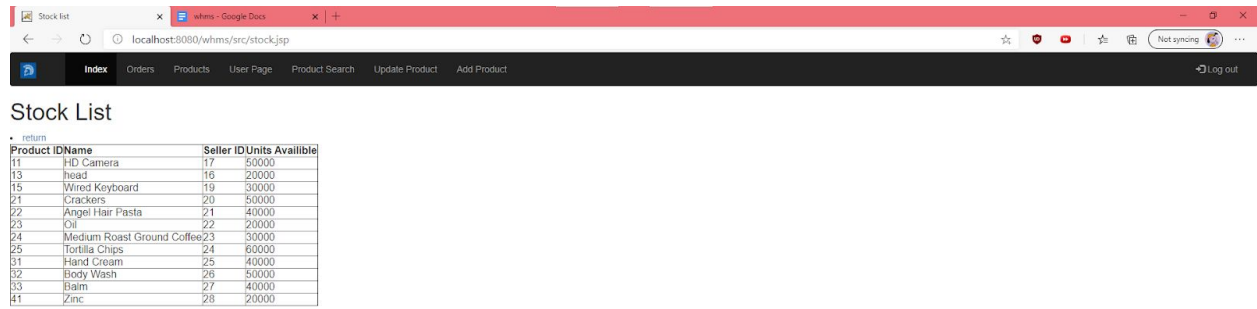
product 1 x

Index page

Our index page helps to navigate to the tracking order page to see the order information and delivery details. Then there is a stock page to see the availability of the product. In the same way, users can navigate to products, add product, user profile, user list, and Generate Bill page.



Stock List

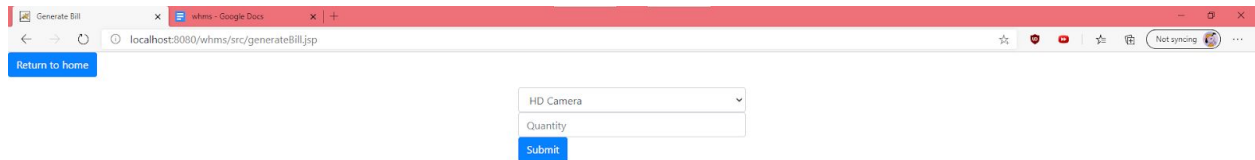


Product ID	Name	Seller ID	Units Available
11	HD Camera	17	50000
13	head	16	20000
15	Wired Keyboard	19	30000
21	Crackers	20	50000
22	Angel Hair Pasta	21	40000
23	Oil	22	20000
24	Medium Roast Ground Coffee	23	30000
25	Tortilla Chips	24	60000
31	Hand Cream	25	40000
32	Body Wash	26	50000
33	Balm	27	40000
41	Zinc	28	20000

Generate Bill

This page will generate the bill by adding the product name with the quantity and it will show order id to the user.

Users must go to the Generate Bill tab on Index. Then they will see a page with all products shown in a dropdown menu. After selecting a product, enter a quantity (minimum 1) that will generate a bill and add it to the database.



Generate Bill

Return to home

HD Camera

Quantity

Submit

localhost:8080/whms/src/generateBill.jsp

After generating a bill, you will get a "success" message in a redirected page. This implies that the bill was successfully added to the database. You will also get an order number, the primary key for the Billing_info table.



Return to homepage

localhost:8080/whms/src/generateBillComplete.jsp

Generating bill complete: Your order ID is:

43

	order_id	user_id	total
	37	6	48
	38	47	26
	39	47	32
	40	54	36
	41	54	4
	42	0	4
	43	6	4
	NULL	NULL	NULL

	order_id	date	time	address_id	address
	37	2020-12-05	23:23:46	6	5,Roseline St,San Jose,CA-95123
	38	2020-12-08	00:16:10	47	j
	39	2020-12-08	00:22:21	47	j
	40	2020-12-09	23:49:41	54	hhh drive
	41	2020-12-09	23:58:26	54	hhh drive
	42	2020-12-10	00:43:49	0	
	43	2020-12-11	21:59:48	6	8888 7th Street, San Jose, CA 95121
	NULL	NULL	NULL	NULL	NULL

The new order is inserted into the database in the tables Tracking_info and Order_info.

Track order

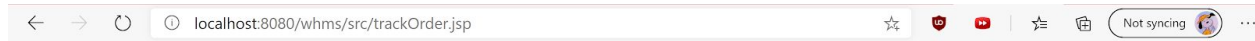
Users must first go to the track order tab on Index.

Users can track their orders by entering order id and it will show the order info, billing info, and tracking information which includes date, time, and address, taken from multiple tables (Order_info, billing_info, and tracking_info).

Track Order

order id

Submit



[Return to track orders](#)

Order Info

Order ID	User ID	Product ID	Category ID	Units	Date
43	6	11	1	1	2020-12-10

Billing

Total Billed (\$)
4

Tracking Info

Order ID	Date	Time	Address ID	Address
43	2020-12-10	18:59:48	6	8888 7th Street, San Jose, CA 95121

User list

We made this page from an administrator point of view that if one wants to check how many sellers and customers are associated with the warehouse, administrators can select by user type and check the list.



Users

Please select user type:

Search selection by name:



Users

• return

User ID	User Name	User Type	Email	Phone Number	Address
1	Mike Wu	customer	mikewu@email.com	1	12 Main St San Jose CA 95125
2	John Smith	customer	johnsmith@email.com	2	Rose St San Jose CA 95125
3	Debbie Abbott	customer	abbottdebbie@email.com	4083333333	10 Rowan St Santa Clara CA 95050
4	Vivek Aggarwal	customer	vivekaggarwal@email.com	4084084444	23 Babylon St San Jose CA 95123
5	Nitin Aggarwal	customer	nitinaggarwal@email.com	4089999988	23 Babylon St San Jose CA 95123
6	Bryan	customer	leslieabert@email.com	1111	111111111
7	Thomas Altura	customer	thomasaltura@email.com	5103454353	5 Roseline St San Jose CA 95123
8	Usha Andra	customer	ushaandra@email.com	5105858585	10 Adobe St San Jose CA 95134
9	Michael Ashley	customer	michaelashley@email.com	5105105100	9 Main St San Jose CA 95125
10	Anu Basu	customer	anubasu@email.com	5109412344	4 Babylon St San Jose CA 95123
11	Michael Beasley	customer	michaob@email.com	5100987678	14 Rowan St Santa Clara CA 95050
12	Roy Blitzer	customer	royb@email.com	5102938475	12 Newbee St San Jose CA 95123
13	Mi Nan Boyd	customer	minanb@email.com	9169394852	21 Rose St San Jose CA 95125
14	Heleen Brubeck	customer	hbrubeck@email.com	9163872435	21 Rose St San Jose CA 95125
15	Joel Busch	customer	joelbusch@email.com	9169161111	76 Main St San Jose CA 95125
43	q	customer	q	q	q
47	j	customer	j	j	j
48	pppp	customer	p@email.com	111	p
49	Mark	customer	Mark@mark.com	4081111111	111 Street st
50	john	customer	email	22	1234 street
51	name	customer	newemail	123	address
52	name	customer	nydkwit	1234	new address
53	jhanvi	customer	jhanvi	1111	59
54	jh	customer	jhjh@email.com	1234567	jhjh drive

User search result



Users

Search results for customer with names containing: Michael

• return

User ID	User Name	User Type	Email	Phone Number	Address
9	Michael Ashley	customer	michaelashley@email.com	5105105100	9 Main St San Jose CA 95125
11	Michael Beasley	customer	michaob@email.com	5100987678	14 Rowan St Santa Clara CA 95050