Name: Samuel Akintomide StudentID: 22041893

Chemicals Website

Description

My application is a website where people can learn about different chemicals as well as buy these chemicals. My audience is everyone. When companies purchase products, they will just need to setup the account as if they were a singular person. Companies have people in the company who will buy the products for them and that's how I have set up my application. In the future the website will also allow for people to sell chemicals through the website, hence why I have left some navigation areas ready for usage such as suppliers etc.

The signup and login form acts as a form of security so that even though anyone can make an account for the website and start purchasing chemicals, there will still be information on them so that we could enquire about reasons for purchasing items and possibly blacklist the person. Chemicals can be extremely dangerous, and purchasers need to be monitored.

Site map (whole structure of site with links)

Home Page (index.xhtml) – Has a search bar to allow people to search for information on chemicals and when the chemical which the user wants information is entered then they will then be brought to a page that stores information about the chemical. This makes it easier to gather information rather than having to blindly search around the page.

Other pages that can be accessed from Home Page:

- Learning Page (learn.xhtml) – Stores information on chemicals so that different users can learn about chemicals. My code allows for each chemical to (in the future) have different pages which will have unique data on them.
- Ordering Page (order.xhtml) – Page where the user can order items. This page allows the users to find out about stock, price and etc. It also allows them to add items to their basket so that they can be purchased
    - Checkout Page (checkout.xhtml) – Allows the user to view their basket which contains items which they would have already added. This page allows them to view all their items which have been added to the basket, delete items which are no longer wanted and checkout.
- Sign Up Page (signup.xhtml) – Allows users to create accounts so that information can be saved and so that they can purchase items.
- Login Page (login.xhtml) – Allows users who have created accounts to re-log in and access information that has been stored and saved.
- Profile Page (profile.xhtml) – Allows the user to see their data and be able to change any information they want (apart from username). The user can also delete their account in this page and all of their data along with it.
    - Delete Page – Form to allow the user to delete their account and all data along with it.
- Contact Page (contact.xhtml) – Used to contact owner (me) for help. Doesn't lead anywhere.

Name: Samuel Akintomide StudentID: 22041893

Usability/design

The Nav bar added to the left of some of the pages helps with navigation. It improves UX (user experience) so that they can locate information and get their easier. For example, the nav bar on the learn page allows the user to scroll through the page with ease and get to where they need to.

The search bar. The search bar on the main page, learn page and order page allows the user to filter information very easily. Rather than being met with large databases in the learn and ordering page the user can simply input one letter into the search box and that will give a more refined set of information that they would want to see.

Nav bar at the top keeps consistency through out the entire site, excluding 3 pages. This makes it easy for a new comer to know how to navigate between the main topics being information, purchasing, and user information editing. The top nav bar (at times works) and changes so that it will display signup and login if a user is not logged in and profile if they are logged in.

Structure of the forms are very natural as well and have a similar structure to other websites. This means that users will have an easy time knowing how to use the pages. In the signup and login form there is an x button located at the top right which will take a user home. Everyone is used to this and it helps with UX.

Explanation of the Java classes

I have four Java files: Chemicals, Users, SearchBean and Information.

First, SearchBean is used on the home page. It affects a search box that will autofill with different suggested outputs as the user types and clicks enter, that would then lead them to different pages in the site. The searchBean stores the different keys which the user can search for such as Hydrogen. Each of these keys are assigned a value being the url for the link to the page. It acts comparatively, so as a user types (and clicks enter) it will take the string which the user has typed and check to see whether and keys contain the string in them. If so then they will be displayed.

User.java connects to an SQL database which stores information on the users such as name, username, phone number etc. It is used in the signup and profile page. It is used to create accounts for users, log them in, update their information, and delete their accounts. It follows the CRUD structure.

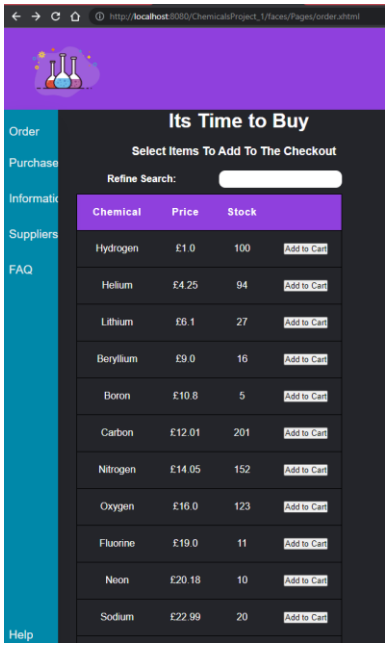
Chemicals also connects to the SQL database and stores information on the products that are being sold such as name, price, stock. This allows the user to see all the information needed when purchasing products and follows the same basic structure of the user page when gathering and sending data. It implements a cart which allows the user to purchase items. The stock values in the table will decrease as a user makes a purchase.

Information also connects to the SQL database and stores information on all Chemicals (that I added to the database) such as name, boiling point, description. This allows the user to see all the information they could need on a chemical. This class allows the user to refine the table displaying the data on the chemicals so that they aren't met with a lot of unimportant information on chemicals they may not want to see. It also allows them to search for specific chemicals rather than having to scan the whole table by themselves.

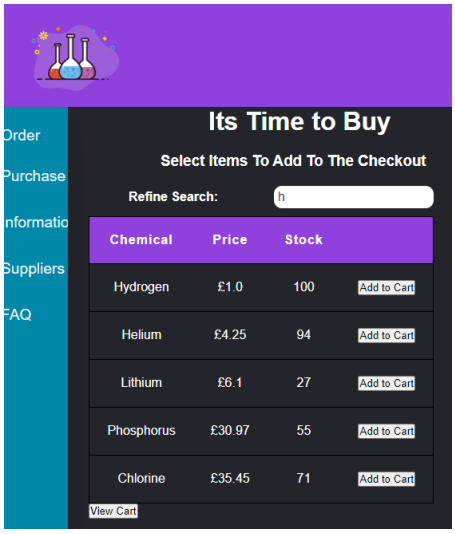
Name: Samuel Akintomide StudentID: 22041893

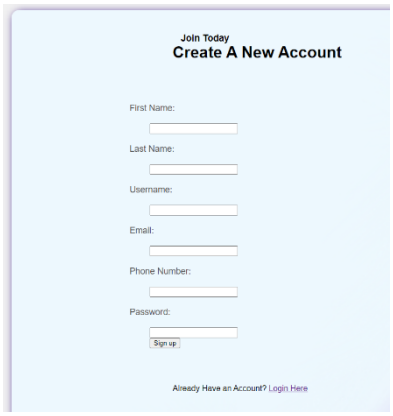
Test plan and test data, screen dumps (live, 2 per page) with annotation

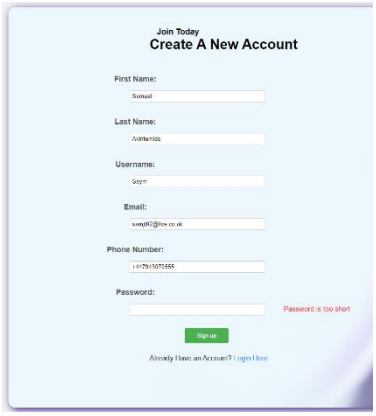
I tested the application as a user and as an admin (I was allowed to look at the SQL spreadsheets). I created fake data for different people so that I could test some features such as validation (making sure passwords are long enough and making sure people can't use the same username, email, or phone number).



In the order page all products are displayed. In a real world scenario, there will be more so it will be harder for people to look through the whole page just to find the multiple products they want to buy. By adding an Ajax search function, the user can input a letter, in this case 'h' and chemicals which contain the letter 'h' are displayed. This helps speed up the buying process.



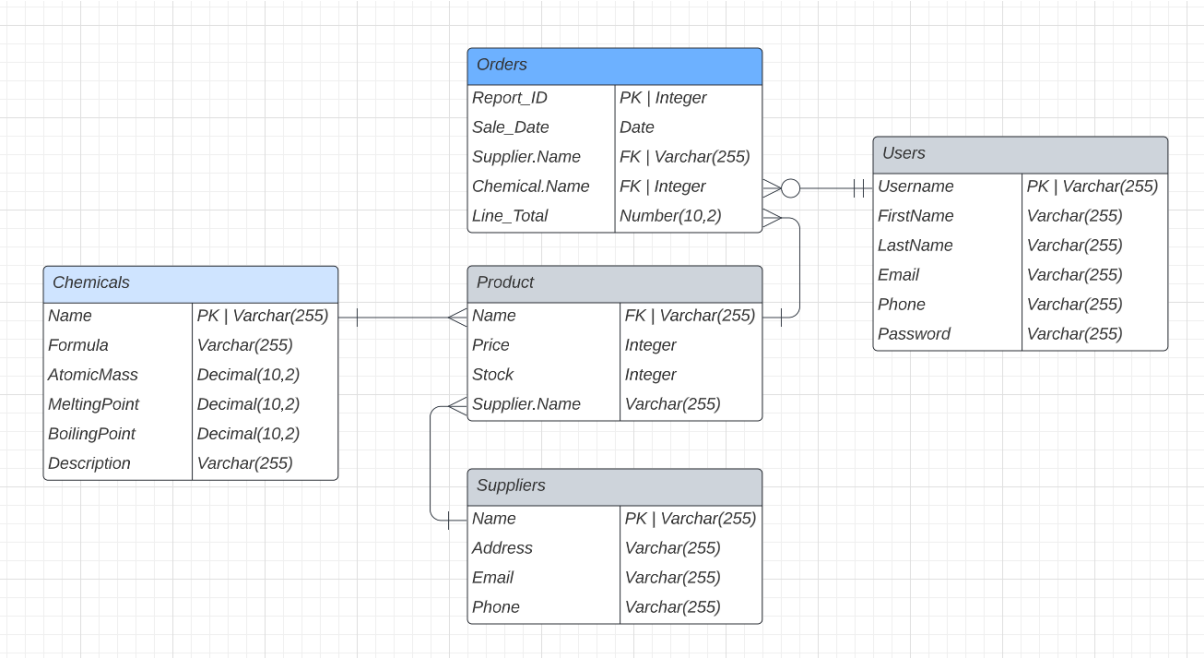




This is a signup form that allows the users to enter all their details so that they can then make and account and be redirected to the login page. Validation also Used. When there are duplicates in usernames, phone numbers, emails or the password is too short like shown then an error message pops up.

Name: Samuel Akintomide StudentID: 22041893

Database table relationship diagram (+ the tables' structures)



The User can make many orders but doesn't have to. Chemicals are the products which are being sold. We need suppliers to provide the chemicals for us so that they can be sold. Multiple products can be sold as the supplier gives a certain amount that can be redistributed.

Sample Data:

CREATE TABLE USERS (FIRSTNAME VARCHAR(255) NOT NULL, LASTNAME VARCHAR(255) NOT NULL, USERNAME VARCHAR(255) NOT NULL, EMAIL VARCHAR(255) NOT NULL, PHONE VARCHAR(255) NOT NULL, PASSWORD VARCHAR(255) NOT NULL, PRIMARY KEY (USERNAME));

INSERT INTO APP.users (FIRSTNAME, LASTNAME, USERNAME, EMAIL, PHONE, PASSWORD) VALUES (Daniel, 'Davis', 'Admin', 'Admin@fake.com', '+1 202-918-2132','Admin',);

INSERT INTO APP.users (FIRSTNAME, LASTNAME, USERNAME, EMAIL, PHONE, PASSWORD) VALUES ('Kanye', 'West', 'Kanye', 'Kanye_West@fake.com', '+1 505-610-8475', 'Donda');

INSERT INTO APP.users (FIRSTNAME, LASTNAME, USERNAME, EMAIL, PHONE, PASSWORD) VALUES ('Drake', 'idk', 'Drake', 'BizzyDrizzy@fake.com', '+44 7703 952756', '123');

INSERT INTO APP.users (FIRSTNAME, LASTNAME, USERNAME, EMAIL, PHONE, PASSWORD) VALUES ('Micheal', 'Jackson', 'MJ', 'Micheal_Jackson@fake.com', '+44 7459 254102', '456');

INSERT INTO APP.users (FIRSTNAME, LASTNAME, USERNAME, EMAIL, PHONE, PASSWORD) VALUES ('Ariana', 'Grande', 'DangerousW', 'Ariana_Grande@fake.com', '+44 7700 152821', '789');

INSERT INTO APP.users (FIRSTNAME, LASTNAME, USERNAME, EMAIL, PHONE, PASSWORD) VALUES ('Micheal', Jordan, 'Creed', 'Micheal_B_Jordan@fake.com', '+44 7911 265898', 'Adonis');

CREATE TABLE PRODUCTS ("NAME" VARCHAR(255) NOT NULL, PRICE DECIMAL(10, 2) NOT NULL, STOCK INTEGER DEFAULT 0 NOT NULL, PRIMARY KEY ("NAME"));
INSERT INTO products (name, price, stock) VALUES ('Hydrogen', 1.00, 100);

Name: Samuel Akintomide StudentID: 22041893

INSERT INTO products (name, price, stock) VALUES ('Helium', 4.25, 94);

INSERT INTO products (name, price, stock) VALUES ('Lithium', 6.10, 27);

INSERT INTO products (name, price, stock) VALUES ('Beryllium', 9.00, 16);

INSERT INTO products (name, price, stock) VALUES ('Boron', 10.80, 5);

INSERT INTO products (name, price, stock) VALUES ('Carbon', 12.01, 201);

INSERT INTO products (name, price, stock) VALUES ('Nitrogen', 14.05, 152);

INSERT INTO products (name, price, stock) VALUES ('Oxygen', 16.00, 123);

INSERT INTO products (name, price, stock) VALUES ('Fluorine', 19.00, 11);

INSERT INTO products (name, price, stock) VALUES ('Neon', 20.18, 10);

INSERT INTO products (name, price, stock) VALUES ('Sodium', 22.99, 20);

INSERT INTO products (name, price, stock) VALUES ('Magnesium', 24.31, 120);

INSERT INTO products (name, price, stock) VALUES ('Aluminum', 26.98, 115);

INSERT INTO products (name, price, stock) VALUES ('Silicon', 28.09, 106);

INSERT INTO products (name, price, stock) VALUES ('Phosphorus', 30.97, 55);

INSERT INTO products (name, price, stock) VALUES ('Sulfur', 32.06, 122);

INSERT INTO products (name, price, stock) VALUES ('Chlorine', 35.45, 71);

INSERT INTO products (name, price, stock) VALUES ('Argon', 39.95, 60);

CREATE TABLE INFORMATION (Chemical VARCHAR(255) NOT NULL, FORMULA VARCHAR(255) NOT NULL, ATOMICMASS DECIMAL(10,2) DEFAULT 0.00 NOT NULL, MELTINGPOINT DECIMAL(10, 2) DEFAULT 0.00 NOT NULL, BOILINGPOINT DECIMAL(10, 2) DEFAULT 0.00 NOT NULL,DESCRIPTION LONG VARCHAR, PRIMARY KEY (Chemical));

INSERT INTO APP.INFORMATION (CHEMICAL, FORMULA, ATOMICMASS, MELTINGPOINT, BOILINGPOINT, DESCRIPTION)

VALUES ('Hydrogen', 'H', 1.01, -259.14, -252.87, 'Hydrogen is the lightest and most abundant element in the universe.');

INSERT INTO APP.INFORMATION (CHEMICAL, FORMULA, ATOMICMASS, MELTINGPOINT, BOILINGPOINT, DESCRIPTION)

VALUES ('Helium', 'He', 4.00, -272.20, -268.93, 'Helium is a colorless, odorless, tasteless, non-toxic, inert monatomic gas that heads the noble gas group in the periodic table.');

INSERT INTO APP.INFORMATION (CHEMICAL, FORMULA, ATOMICMASS, MELTINGPOINT, BOILINGPOINT, DESCRIPTION)

VALUES ('Lithium', 'Li', 6.94, 180.54, 1342.00, 'Lithium is a soft, silvery-white alkali metal.');

Name: Samuel Akintomide StudentID: 22041893

INSERT INTO APP.INFORMATION (CHEMICAL, FORMULA, ATOMICMASS, MELTINGPOINT, BOILINGPOINT, DESCRIPTION)

VALUES ('Beryllium', 'Be', 9.01, 1278.00, 2970.00, 'Beryllium is a relatively rare element in the universe, usually occurring as a product of the spallation of larger atomic nuclei.');

INSERT INTO APP.INFORMATION (CHEMICAL, FORMULA, ATOMICMASS, MELTINGPOINT, BOILINGPOINT, DESCRIPTION)

VALUES ('Boron', 'B', 10.81, 2349.00, 4200.00, 'Boron is a low-abundance element in the Solar system and in the Earth's crust.');

INSERT INTO APP.INFORMATION (CHEMICAL, FORMULA, ATOMICMASS, MELTINGPOINT, BOILINGPOINT, DESCRIPTION)

VALUES ('Carbon', 'C', 12.01, 3500.00, 4827.00, 'Carbon is a non-metallic element and the fourth most abundant element in the universe by mass, after hydrogen, helium, and oxygen.');

INSERT INTO APP.INFORMATION (CHEMICAL, FORMULA, ATOMICMASS, MELTINGPOINT, BOILINGPOINT, DESCRIPTION)

VALUES ('Nitrogen', 'N', 14.01, -210.00, -195.79, 'Nitrogen is a colorless, odorless, and tasteless gas that makes up about 78% of the Earth's atmosphere.');

INSERT INTO APP.INFORMATION (CHEMICAL, FORMULA, ATOMICMASS, MELTINGPOINT, BOILINGPOINT, DESCRIPTION)

VALUES ('Oxygen', 'O', 16.00, -218.79, -182.96, 'Oxygen is a colorless, odorless, and tasteless gas that is essential for most life forms on Earth.');

INSERT INTO APP.INFORMATION (CHEMICAL, FORMULA, ATOMICMASS, MELTINGPOINT, BOILINGPOINT, DESCRIPTION)

VALUES ('Fluorine', 'F', 19.00, -219.67, -188.11, 'Fluorine is a pale yellow, highly reactive gas that is the most electronegative element.');

INSERT INTO APP.INFORMATION (CHEMICAL, FORMULA, ATOMICMASS, MELTINGPOINT, BOILINGPOINT, DESCRIPTION)

VALUES ('Neon', 'Ne', 20.18, -248.59, -246.08, 'Neon is a colorless, odorless, inert gas that is the second lightest noble gas.');

INSERT INTO APP.INFORMATION (CHEMICAL, FORMULA, ATOMICMASS, MELTINGPOINT, BOILINGPOINT, DESCRIPTION)

VALUES ('Sodium', 'Na', 22.99, 97.80, 883.00, 'Sodium is a soft, silvery-white metal that is highly reactive and can ignite spontaneously in air.');

INSERT INTO APP.INFORMATION (CHEMICAL, FORMULA, ATOMICMASS, MELTINGPOINT, BOILINGPOINT, DESCRIPTION)

VALUES ('Magnesium', 'Mg', 24.31, 650.00, 1090.00, 'Magnesium is a silvery-white, light metal that is relatively abundant in the Earth's crust.');

Name: Samuel Akintomide StudentID: 22041893

A reflection of your experience

I honestly found the whole process of making the website very difficult. Things were very tedious, and a lot of research had to be done to understand what was being done and how to fix any issues that I ran into. While that may not have been a bad thing, it made me feel lose motivation to do the work as I was always running into issues.

This has helped me when it comes to troubleshooting and coding, but I did find this project quite difficult. I have to say that setting up the Ajax search form bar at the start is probably one of the hardest things to do that took a lot of research alongside deleting data when the user requested to do so. For some reason these two ran into a lot of bugs. Even now, in my checkout page I cant remove items from the basket on that page but if I were to do it on the order page then it would work. I have come to the conclusion that I cant live edit a table i.e be on the same page as a table that needs to refresh and displays live values.

The project was fun at times. Being able to design applications like this that actually work is a very nice feeling.