

# Cheers!

Keith Bryan

@braincandle76

<https://github.com/braincandle76/cheers>

# Description

Cheers is a drink rating app that allows users to submit and rate a variety of drinks. It allows people to show the most rated drinks at the time to show what's popular. Users can create an account, submit a drink, give a rating, and look up others drinks that were already submitted and give a rating as well. Non-registered users can look up a specific drink and view the rating. This allows users an ability to either share their opinion of a drink they already had or find something new they've never tried.



# Features

- Users can create an account.
- Registered users can submit, rate, search and view drinks.
- Non-Registered users can view drinks and search drinks.



# Planning - User Stories

**As a viewer I can search and view new drinks.** This entire project is built on having viewers to look and see what drinks are currently popular and find new drinks.

**As a registered user I can submit drinks that aren't in the system and rate them.** For this list to exist, viewers create an account to register drinks that they have had and give a rating of that drink.



# Planning - Database

I have three tables in my database -- Users, Drinks, and Rating. All three have a primary key (id). The Rating table is linked to both the Users table and Drinks table.

Making the Rating table having a one-to-one relationship between the Users and the Drinks.

When a user rates a drink, the User\_ID is linked to that Drink\_ID with the rating of that said drink.



# Technology Stack

- Spring Boot
- Java
- MySQL
- Thymeleaf
- Bootstrap



# Demo - Create User 1

Please fill in this form to create an account.

First :

Last :

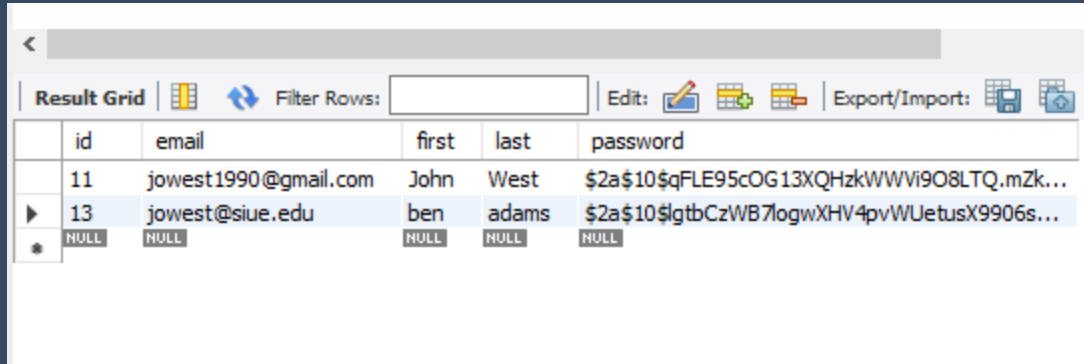
Email id :

Password :

By creating an account you agree to our [Terms & Privacy](#).



# Demo - Create User 2



The screenshot shows a web application interface with a 'Result Grid' table. The table has columns for 'id', 'email', 'first', 'last', and 'password'. The first row shows a user with id 11, email jowest1990@gmail.com, first name John, last name West, and a long password. The second row shows a user with id 13, email jowest@siue.edu, first name ben, last name adams, and another long password. The third row shows NULL values for all columns. The interface includes a search bar, a 'Filter Rows' button, and an 'Edit' button.

	id	email	first	last	password
	11	jowest1990@gmail.com	John	West	\$2a\$10\$qFLE95cOG13XQHkzWWVi9O8LTQ.mZk...
▶	13	jowest@siue.edu	ben	adams	\$2a\$10\$gtbCzWB7logwXHV4pvWUetusX9906s...
*	NULL	NULL	NULL	NULL	NULL





# Demo Code - Create User

```
    @PostMapping("/process_register")  
    public String processRegister(User user) {  
        userRepo.save(user);  
        return index();  
    }  
    /**
```



# Demo – Create drink 1

[User Page](#) [New Drink](#) [Logout](#)

## Activity

Drink Name

Brand

Is it  
Carbonated?

Is it  
Alcoholic?

Type of Drink



# Demo – Create drink 2

Result Grid						
				Filter Rows:		Edit:
	id	alcoholic	brand	carbonated	name	type
▶	18	0	Budweiser	0	bud light	
	19	0	coca cola	1	coke	Soda
	20	0	hi-ci	0	orange juice	Juice
	21	1	miller	1	blue moon	Beer
	22	0	coca cola	1	sprite	Soda
	23	1	Paulaner	1	Okttoberfes...	Beer
*	NULL	NULL	NULL	NULL	NULL	NULL



# Demo code – Create drink

```
public void setId(int id) { this.id = id; }

public String getName() { return name; }

public void setName(String name) { this.name = name; }

public String getBrand() { return brand; }

public void setBrand(String brand) { this.brand = brand; }

public boolean isCarbonated() { return carbonated; }

public void setCarbonated(boolean carbonated) { this.carbonated = carbonated; }

public boolean isAlcoholic() { return alcoholic; }

public void setAlcoholic(boolean alcoholic) { this.alcoholic = alcoholic; }

public String getType() { return type; }

public void setType(String type) { this.type = type; }

public String getRating() { return rating; }

public void setRating(String rating) { this.rating = rating; }
```



# Demo – login 1

Please sign in

jowest1990@gmail.com

.....|

Sign in



# Demo – login 2



Cheers**John West!**

[User Page](#)[New Drink](#)[Logout](#)[Preferences](#)

## Activity

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Last updated Mon Sep 20 2021 12:13:15 GMT-0500 (Central Daylight Time)

## Another Section

Some text.



# Demo code- login

```
@Override
protected void configure(HttpSecurity http) throws Exception {
    http.authorizeRequests() ExpressionUrlAuthorizationConfigurer<...>.ExpressionInterceptUrlRegistry
        .antMatchers( ...antPatterns: "/users").authenticated()
        .anyRequest().permitAll()
        .and() HttpSecurity
        .formLogin() FormLoginConfigurer<HttpSecurity>
            .usernameParameter("email")
            .defaultSuccessUrl("/users")
            .permitAll()
        .and() HttpSecurity
        .logout().logoutSuccessUrl("/").permitAll();
}
```



# Demo - login

The code from the previous slide handles the user authentication.

- The `configure()` method; when a user selects a link or button containing `@{/login}`, a login form is generated by Spring Boot Security package for the user to enter their email and password.
- Once entered, the system will search the user by email and to authenticate, it matches the password that is saved in the DB.
- If successful, the user is redirected to the `/users` page and can logout at any moment the user pleases





# What I Learned

- I learned about building a project from scratch, from the brainstorming phase to actually writing code. It took a lot of planning to come up with a realistic and useable web application within the given timeframe.
- This was my first time implementing the Agile framework for software development in a practical application. Our group utilized ZenHub to track progress and we created pull requests to a GitHub repository where we each had our own development branch outside of the main branch.
- I continued to work on my knowledge and understanding of technologies that we utilized (Java, MySQL, thymeleaf, boot strap) in order to deliver a polished product.



# What's Next

- To organize drinks by rating
- To view drinks for non-registered users.
- To filter out a specific drink or a group of drinks.

