

Current Weather App

Description

This app is built using Java Spring Boot, Hibernate, MySQL DB and HTML5 (Thymeleaf), Bootstrap, CSS. It can be used to check real time weather and also store the information in the database.

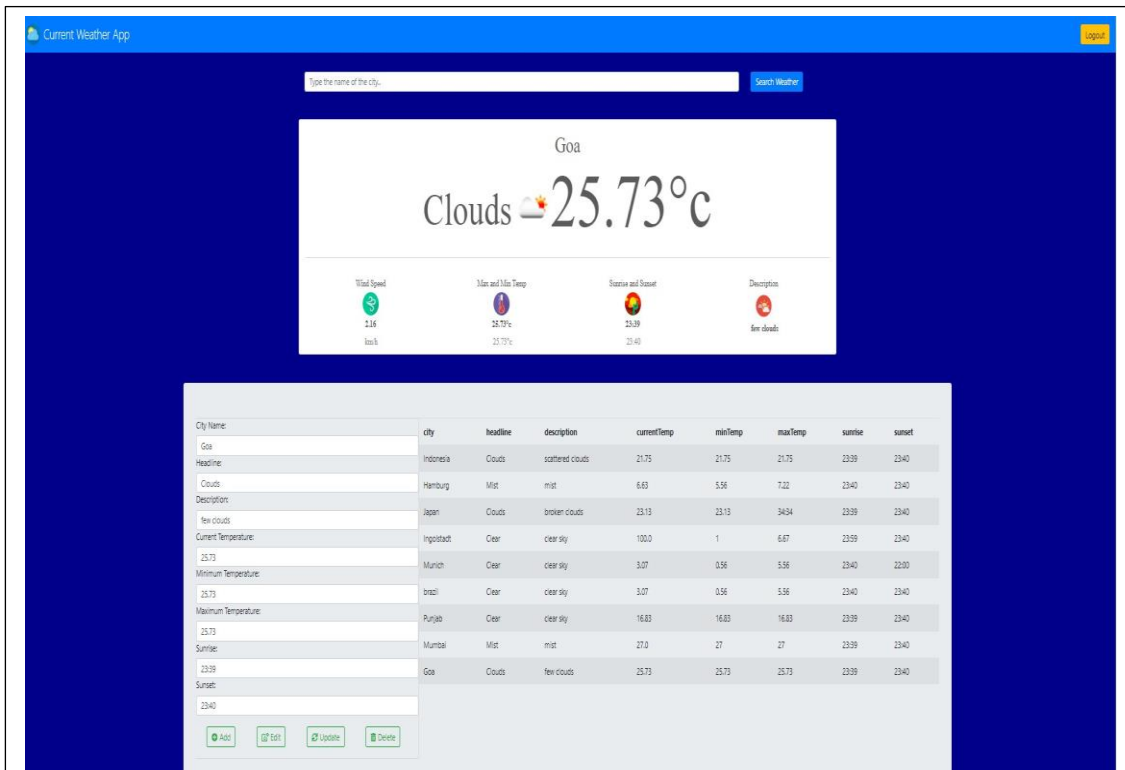


Figure 1: Current Weather Page

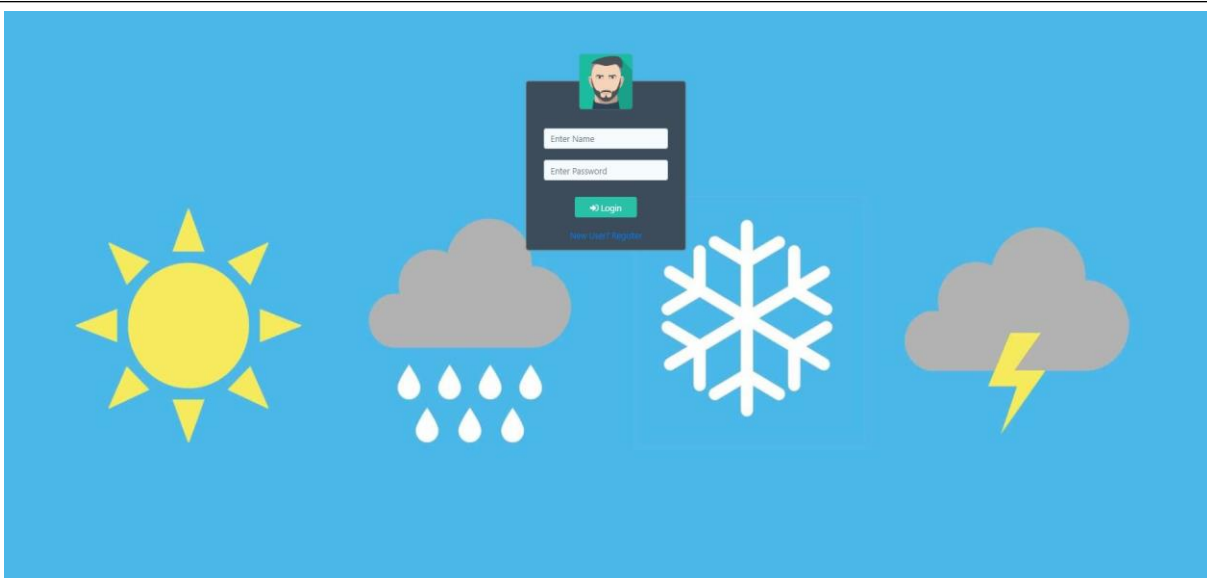


Figure 2: Login Page

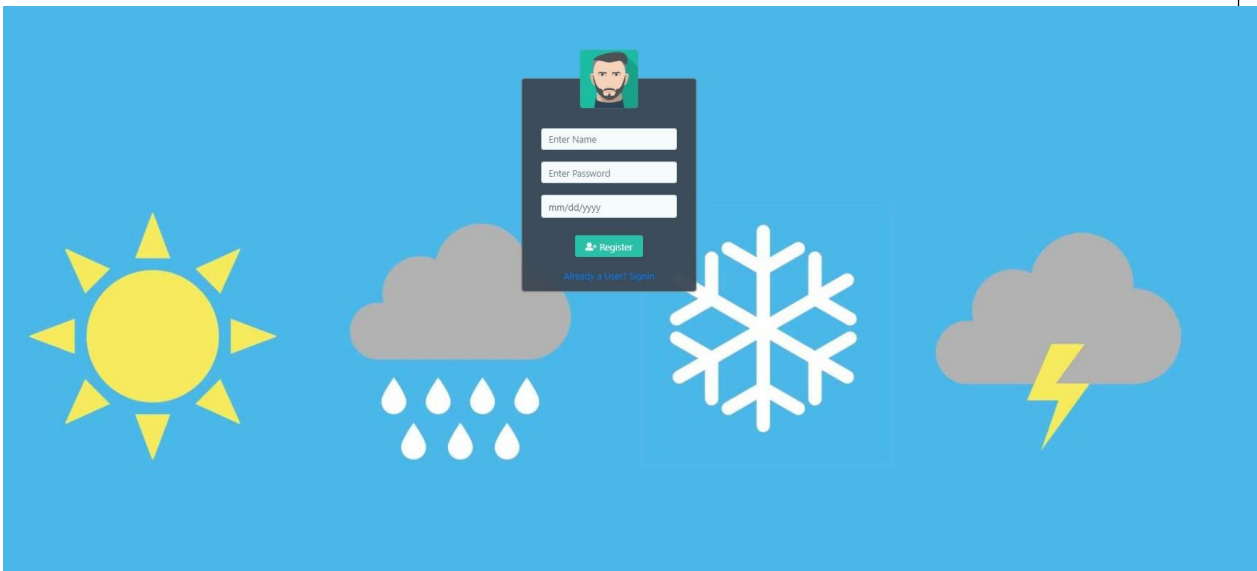


Figure 3: Register Page

Functionalities Implemented

- User Related
 - User Registration (username, password and date of birth)
 - User Login (username and password)
 - Validation of user inputs
- Weather Related
 - Search for weather based on City name located anywhere in the world
 - History table
 - CRUD operations buttons for edit, update, delete and adding city
- Database
 - One table for weather with City Name, Weather Description, Current Temperature, Min Temperature, Max Temperature, Sunrise and Sunset columns
Unique constraint: City name
 - Second table for user details with Username, Password and Date of Birth columns
Unique constraint: Combined on Username and Date of Birth

id	city	currentTemp	descriptions	headline	maxTemp	minTemp	sunrise	sunset
17	Indonesia	21.75	scattered clouds	Clouds	21.75	21.75	23:39	23:40
18	Hamburg	6.63	mist	Mist	7.22	5.56	23:40	23:40
19	Japan	23.13	broken clouds	Clouds	34:34	23.13	23:39	23:40
20	Ingolstadt	100	clear sky	Clear	6.67	1	23:59	23:40
21	Munich	3.07	clear sky	Clear	5.56	0.56	23:40	22:00
25	brazil	3.07	clear sky	Clear	5.56	0.56	23:40	23:40
26	Punjab	16.83	clear sky	Clear	16.83	16.83	23:39	23:40
27	Mumbai	27	mist	Mist	27	27	23:39	23:40
28	Goa	25.73	few clouds	Clouds	25.73	25.73	23:39	23:40

Figure 4: Weather Condition Database Table

	id	pass	date_of_birth	name
1	1	1234	2019-03-19	dhruvjit
2	5	1234	2019-03-14	person

Figure 5: User Details Database Table

- CRUD Operations and History Table:
 - After searching for city name, user can also add it in database. User can even edit, delete and update the details if he wishes to change it in database

city	headline	description	currentTemp	minTemp	maxTemp	sunrise	sunset
Hamburg	Clear	clear sky	17.08	15.56	18.33	12:38 AM	12:39 AM
Munich	Clear	clear sky	15.59	13	18.33	12:38 AM	12:39 AM
China	Clear	clear sky	18.19	18.19	18.19	12:39 AM	12:39 AM
Japan	Rain	light rain	22.79	22.79	22.79	12:38 AM	12:39 AM
Indonesia	Rain	light rain	22.79	22.79	22.79	12:38 AM	12:39 AM

- Validation
 - Validations are configured on login and registration screen
 - Validation is also configured on three fields of the history table i.e. Current Temperature, Sunrise and Sunset. An e.g. is depicted below as per requirements

Scattered clouds

Current Temperature:

length for current temperature should be less than 100

Minimum Temperature:

Maximum Temperature:

Sunrise:

Incorrect! only HH:MM allowed where H->[0-23] and M->[0-59]

Sunset:

Incorrect! only HH:MM allowed where H->[0-23] and M->[0-59]

- The webpage will also redirect to not found page if the city name is incorrect



- Logout
 - A logout screen is also implemented for user if we wishes to end the session



Functionalities Not Implemented:

Considering the Time constraint below mentioned things were not implemented:

- Future development task: Database table to have relations to do the search based on User and city
- Junit testing: I wished to use Mockito for my purpose and however due to occupied schedule this last week I spent too many hours understanding the framework and trying it out, and I couldn't get it to work for my implemented services in time. I have still left some unfinished code in the test repo of the project. However I got to learn a lot regarding testing with this task which I look forward to try out on other projects.

Bugs:

- Tried a lot to end browser session after logout so that user cannot press back button to go to same page. But the solutions I tried don't work due to strange reasons. I tried many different techniques but still this bug lets you to go back on previous page by just pressing back button after every action e.g. login, logout n such (password and user details are not cached only weather related pages)