

=====

02-Mini Project

=====

Requirement : User management

- 1) User Registration
- 2) Unlock Account
- 3) Login Functionality
- 4) Dashboard

Third Party API : <https://dummyjson.com/quotes/random>

=====

Database tables

=====

COUNTRY_MASTER

- COUNTRY_ID INT PK AUTO_INCREMENT
- COUNTRY_NAME VARCHAR

STATES_MASTER

- STATE_ID INT PK AUTO_INCREMENT
- STATE_NAME VARCHAR
- COUNTRY_ID INT FK

CITIES_MASTER

- CITY_ID INT PK AUTO_INCREMENT
- CITY_NAME VARCHAR
- STATE_ID INT FK

USER_MASTER

- USER_ID INT PK AUTO_INCREMENT
- USERNAME VARCHAR
- EMAIL VARCHAR
- PHNO INT
- PWD VARCHAR
- COUNTRY_ID INT FK
- STATE_ID INT FK
- CITY_ID INT FK
- PWD_RESET VARCHAR
- CREATED_DATE TIMESTAMP
- UPDATED_DATE TIMESTAMP

=====

Java & UI Components

=====

Entity & Repository

- UserEntity.java & UserRepo.java
- CountryEntity.java & CountryRepo.java
- StateEntity.java & StateRepo.java

- CityEntity.java & CityRepo.java

Bindings / DTO / Pojos

- RegisterFormDTO.java
- LoginFormDTO.java
- ResetPwdFormDTO.java
- QuoteApiResponseDTO.java
- UserDTO.java

Services

- UserService.java

```
public Map<Integer,String> getCountries();  
  
public Map<Integer,String> getStates(Integer countryId);  
  
public Map<Integer,String> getCities(Integer stateId);  
  
public boolean duplicateEmailCheck(String email);  
  
public boolean saveUser(RegisterFormDTO regFormDTO);  
  
public UserDTO login(LoginFormDTO loginFormDTO);  
  
public boolean resetPwd(ResetPwdFormDTO resetPwdDTO);  
  
public UserDTO getUserByEmail(String email);
```

- DashboardService.java

```
public QuoteApiResponseDTO getQuote();
```

- EmailService.java

```
public boolean sendEmail(String subject, String body, String to);
```

Controllers

- UserController.java
- DashboardController.java

Views

- login.html
- register.html
- resetPwd.html
- dashboard.html
- app.js

```
=====  
Development Process  
=====
```

1) Create spring-boot app with below dependencies

- a) web-starter
- b) thymeleaf
- c) data-jpa
- d) mysql-driver

- e) mail-starter
- f) devtools
- g) jquery

Note: JQuery dependency is not available in STS ide directly hence we need to add it manually.

```
<dependency>
  <groupId>org.webjars</groupId>
  <artifactId>jquery</artifactId>
  <version>3.6.4</version>
</dependency>
```

2) Create Entities and Repositories

3) Create Binding or DTO classes

4) Create Service interfaces with required methods

5) Create Controller classes

6) Create View Pages

7) Configure below properties in application.properties file

- a) datasource
- b) orm
- c) smtp

=====

```
spring.application.name=02_user_mgmt_app
```

#Data Source Properties

```
spring.datasource.url=jdbc:mysql://localhost:3306/jrtp27
spring.datasource.username=root
spring.datasource.password=root
```

#ORM Properties

```
spring.jpa.hibernate.ddl-auto=update
spring.jpa.show-sql=true
spring.jpa.properties.hibernate.format_sql=true
```

#SMTP Properties

```
spring.mail.host=smtp.gmail.com
spring.mail.port=587
spring.mail.username=<email>
spring.mail.password=<gmail-app-pwd>
spring.mail.properties.mail.smtp.auth=true
spring.mail.properties.mail.smtp.starttls.enable=true
spring.mail.properties.mail.smtp.starttls.required=true
```

=====

=====

AJAX Introduction

=====

1) Synchronous request (entire web page will be reloaded)

2) Asynchronous request (page will not be reloaded, only data will be re-loaded)

in our app, for states and cities drop downs we need to implement asynchronous requests.

=> To implement asynchronous request we will use AJAX.

-> To start AJAX engine we can use JQuery library

Step-1 : Add below dependency in pom.xml

```
<dependency>
    <groupId>org.webjars</groupId>
    <artifactId>jquery</artifactId>
    <version>3.6.4</version>
</dependency>
```

Step-2 : include js file in html page <head/> section

```
<script src="/webjars/jquery/3.6.4/jquery.min.js"></script>
```

step-3 : implement ajax calls to load drop down options dynamically

```
<script>
```

```
$(document).ready(function () {

    $("#countryId").on("change", function () {
        var cid = $("#countryId").val();

        $('#stateId').find('option').remove();
        $('<option>').val('').text('-Select-').appendTo("#stateId");

        $('#cityId').find('option').remove();
        $('<option>').val('').text('-Select-').appendTo("#cityId");

        $.ajax({
            type: 'GET',
            url: '/states/' + cid,
            success: function (response) {
                // iterate response entries and display as state drop down options
                $.each(response, function (key, value) {
                    $('<option>').val(key).text(value).appendTo("#stateId");
                });
            }
        });
    });

    $("#stateId").on("change", function () {
        var stateId = $("#stateId").val();

        $('#cityId').find('option').remove();
        $('<option>').val('').text('-Select-').appendTo("#cityId");

        $.ajax({
            type: 'GET',
            url: '/cities/' + stateId,
            success: function (response) {
                // iterate response entries and display as state drop down options
                $.each(response, function (key, value) {
                    $('<option>').val(key).text(value).appendTo("#cityId");
                });
            }
        });
    });
});
```

```
});  
</script>
```