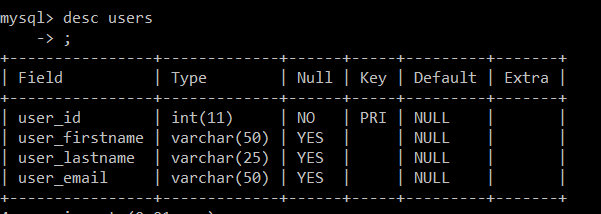
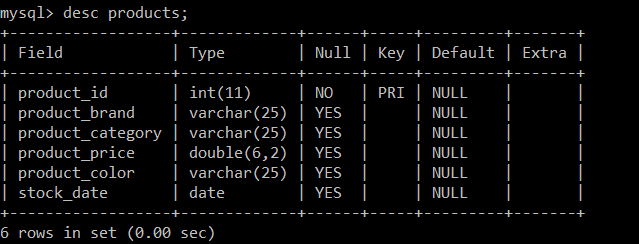
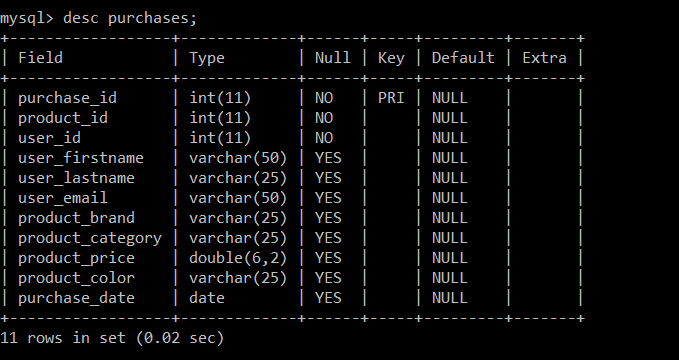
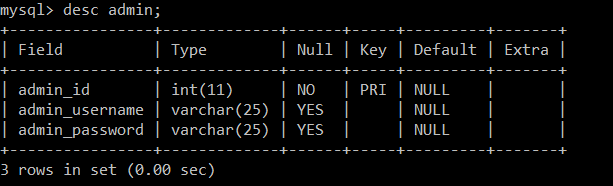
PG FSD Implement OOPS using JAVA with Data Structures and Beyond

Phase-4 Project Specification Document

* Introduction

This is a spring-boot application which allows admin user to login and change password if needed. Also admin user has the ability to manage the information about products, users, and purchase records.

* Database schema
  + Users - to store user information  
    
  + Products - to store product information  
    
  + Purchases - to store purchase records  
    
  + Admin - to store admin users’ username and password  
    
* APIs
  + Admin
    - *Login with username and password*
      * POST <http://localhost:8080/admin/login>
    - *Change password*
      * POST <http://localhost:8080/admin/changePassword>
  + Product
    - *Get all products*
      * GET <http://localhost:8080/products/all>
    - *Add a new product*
      * POST <http://localhost:8080/products/add>
    - Search product by id
      * GET http://localhost:8080/products/product/{id}
    - *Update a product information*
      * PUT [http://localhost:8080/products/update/{id}](http://localhost:8080/products/all)
    - *Delete a product*
      * DELETE [http://localhost:8080/products/delete/{id}](http://localhost:8080/products/all)
    - *Search products by field*
      * POST [http://localhost:8080/products/searchByField](http://localhost:8080/products/all)
  + User
    - Get all users
      * GET <http://localhost:8080/users/all>
    - Add a new user
      * POST [http://localhost:8080/users/add](http://localhost:8080/users/all)
    - Search user by id
      * GET [http://localhost:8080/users/user/{id}](http://localhost:8080/users/all)
    - Update user information
      * PUT [http://localhost:8080/users/update/{id}](http://localhost:8080/users/all)
    - Delete a user
      * DELETE [http://localhost:8080/users/delete/{id}](http://localhost:8080/users/all)
    - Search users by field
      * POST [http://localhost:8080/users/searchByField](http://localhost:8080/users/all)
  + Purchase
    - Get all purchase records
      * GET <http://localhost:8080/purchases/all>
    - Search purchase records by field
      * POST [http://localhost:8080/purchases/searchByField](http://localhost:8080/purchases/all)
* Usage of application
  + Run as Maven Clean
  + Run as Maven install
  + Run as Spring Application
  + For detailed payload information about APIs please refer to the screenshots.