**Project Name :- STYLEASY : RENT WEAR SWAP**

**Project Member :-**

**Rina Panchbhai 220343120089**

**Rushikesh Atalkar 220343120092**

**Shruti Patil 220343120073**

**Gaurav Kalal 220343120034**

**Abstract :-**

This project is about renting the clothes. This project will solve the problem of buying expensive clothes and not been used later. It will include clothes for all men, women and kids.

It provides the user with a catalogue of different category of clothes available for renting from the website. This idea of the app is actually really cool since nowadays, with all the Instagram fast fashion many people just need an outfit for one photoshoot and then they will never wear it again.

In order to develop an cloth rental website, a number of Technologies must be studied and understood. These include multi-tiered architecture, server and client side scripting techniques(React JS), implementation technologies such as Spring Boot, programming language (Core Java, Advance Java), relational databases (MySQL).

This is a project with the objective to develop a basic website where a consumer is provided with a cloth rental facility. Where the user will be able to give his clothes also for rent. This is a great way to save money on clothes that you only wear once.

**Implementation Technologies :-**

1. **Spring Boot:**

Spring Boot makes it easy to create stand-alone, production-grade application and provides the RAD (Rapid Application Development) feature.

We take an opinionated view of the Spring platform and third-party libraries so you can get started with minimum fuss. Most Spring Boot applications need minimal Spring configuration.

**1.1 Features of Spring Boot :-**

**Web Development:**

It is a well-suited Spring module for web application development. We can easily create a self-contained HTTP application that uses embedded servers like **Tomcat, Jetty,** or Undertow. We can use the **spring-boot-starter-web** module to start and run the application quickly.

**Spring Application:**

The SpringApplication is a class that provides a convenient way to bootstrap a Spring application. It can be started from the main method.

**Application Events and Listeners:**

Spring Boot uses events to handle the variety of tasks. It allows us to create factories file that is used to add listeners. We can refer it to using the **ApplicationListener key**.

Always create factories file in META-INF folder like **META-INF/spring.factories**.

**Admin Support:**

Spring Boot provides the facility to enable admin-related features for the application. It is used to access and manage applications remotely. We can enable it in the Spring Boot application by using **spring.application.admin.enabled** property.

**Externalized Configuration:**

Spring Boot allows us to externalize our configuration so that we can work with the same application in different environments. The application uses YAML files to externalize configuration.

**Properties Files:**

Spring Boot provides a rich set of **Application Properties**. So, we can use that in the properties file of our project. The properties file is used to set properties like **server-port =8082** and many others. It helps to organize application properties.

**YAML Support:**

It provides a convenient way of specifying the hierarchical configuration. It is a superset of JSON. The SpringApplication class automatically supports YAML. It is an alternative of properties file.

**Type-safe Configuration:**

The strong type-safe configuration is provided to govern and validate the configuration of the application. Application configuration is always a crucial task which should be type-safe. We can also use annotation provided by this library.

**Logging:**

Spring Boot uses Common logging for all internal logging. Logging dependencies are managed by default. We should not change logging dependencies if no customization is needed.

**Security:**

Spring Boot applications are spring bases web applications. So, it is secure by default with basic authentication on all HTTP endpoints. A rich set of Endpoints is available to develop a secure Spring Boot application.

**1.2 Advantages of Spring Framework :-**

**Spring Boot works well with several servlet containers:**

Spring Boot works well with some of the most popular embedded servlet containers. Spring Boot uses [Tomcat](https://www.adservio.fr/post/tomcat-performance-best-practices) as its default, but you can easily swap it for Jetty, Undertow, Resin, and Wildfly. You get to choose the option that improves the specific types of functionality that concern you most.

Just as importantly, Spring Boot automatically identifies the servlet you set as the new default during the boot sequence. These Advantages of Spring Boot give you the flexibility to choose embedded servers that suit your needs best.

**Bootstrapping saves memory space:**

Spring Boot uses Boot Initializer to compile the source language. This bootstrapping technique makes it possible for users to save space on their devices and load applications quickly.

**Decreased boilerplate code:**

Spring Boot’s in-memory database and embedded server (Tomcat) decrease or eliminate the boilerplate code typically needed to set up an application.

Without lots of boilerplate code, development teams can shorten their development times and update cycles, leading to more satisfied users and more productive employees. It’s yet another of the advantages of Spring Boot that helps developers save time.

**No XML configuration required:**

Spring project developers can choose to use annotations or XML configurations. The option to avoid XML configurations appeals to a lot of software engineers who don’t want to go through the extra steps required.

**WAR files are not required:**

While Spring Boot can use WAR (web application resource) files, they are not necessary. Instead, Spring Boot can rely on JAR (Java resource).

JAR has a shorter, simpler structure that makes them useful for developers and users. The lightweight files work quickly to connect applications with the tools they need to function.

The option to use either WAR or JAR also benefits development teams. If someone in the group doesn’t have experience with JAR, they can rely on WAR. It might have a subtle effect on speed, but it helps developers bring their products to market as fast as possible.

**POM dependency management:**

Spring Boot doesn’t force you to use a parent POM (project object model). Adding the spring-boot-dependencies artifact lets you manage dependencies without relying on a parent POM or XML file.

**A large community of helpful users:**

Like many open-source tools, Spring Boot has a large community of users full of people who enjoy sharing their insights and creations.

No matter what level of experience you have with Spring’s ecosystem of products, you can find helpful tutorials and discussions online.

Another advantage of having such a large community of users is that you can often find existing code that closely resembles what you want to build.

Instead of starting from scratch, you can access the code and adjust it to meet your needs. [GitHub’s Spring Boot page](https://github.com/spring-projects/spring-boot" \t "_blank) is always a good place to start.

1. **The JDBC Template**

The central class of the Spring JDBC abstraction framework is the **JdbcTemplate** class that includes the most common logic in using the JDBC API to access data, such as handling the creation of connection, statement creation, statement execution, and release of resource. The**Jdbc-Template**class can be found in the **org.springframework.jdbc.core**package.

The **JdbcTemplate** class instances are thread-safe once configured. A single **JdbcTemplate** can be configured and injected into multiple DAOs.

We can use the **JdbcTemplate** to execute the different types of SQL statements. **Data Manipulation Language** (**DML**) is used for inserting, retrieving, updating, and deleting the data in the database such as **SELECT**, **INSERT**, or **UPDATE** statements

**2.1** **MySQL:**

MySQL, the most popular Open Source SQL database management system, is developed, distributed, and supported by Oracle Corporation.

**Features of MySQL:**

* **MySQL is a database management system.**

A database is a structured collection of data. It may be anything from a simple shopping list to a picture gallery or the vast amounts of information in a corporate network. To add, access, and process data stored in a computer database, you need a database management system such as MySQL Server. Since computers are very good at handling large amounts of data, database management systems play a central role in computing, as standalone utilities, or as parts of other applications.

* **MySQL databases are relational.**

A relational database stores data in separate tables rather than putting all the data in one big storeroom. The database structures are organized into physical files optimized for speed. The logical model, with objects such as databases, tables, views, rows, and columns, offers a flexible programming environment.

* **MySQL software is Open Source.**

Open Source means that it is possible for anyone to use and modify the software. Anybody can download the MySQL software from the Internet and use it without paying anything.

* **The MySQL Database Server is very fast, reliable, scalable, and easy to use.**

MySQL Server was originally developed to handle large databases much faster than existing solutions and has been successfully used in highly demanding production environments for several years. Although under constant development, MySQL Server today offers a rich and useful set of functions. Its connectivity, speed, and security make MySQL Server highly suited for accessing databases on the Internet.

* **MySQL Server works in client/server or embedded systems.**

The MySQL Database Software is a client/server system that consists of a multithreaded SQL server that supports different back ends, several different client programs and libraries, administrative tools, and a wide range of application programming interfaces (APIs).

1. **Hardware and Software Requirements (Minimum):-**

**Hardware:**

* Intel i3 processor 5th generation or later / AMD Ryzen 200 2nd generation or later
* 2 GB ddr3 ram.
* Windows 7 Home edition or later.
* 200 GB Sata HDD Space.
* Data Connection 200 kbps.

**Software:**

* Eclipse 4.7 Oxygen
* MySQL 5.7 with Workbench 8.0
* Google Chrome version 79.0
* Apache Tomcat Server 8.5
* Maven Dependencies

1. **ER Diagram:**

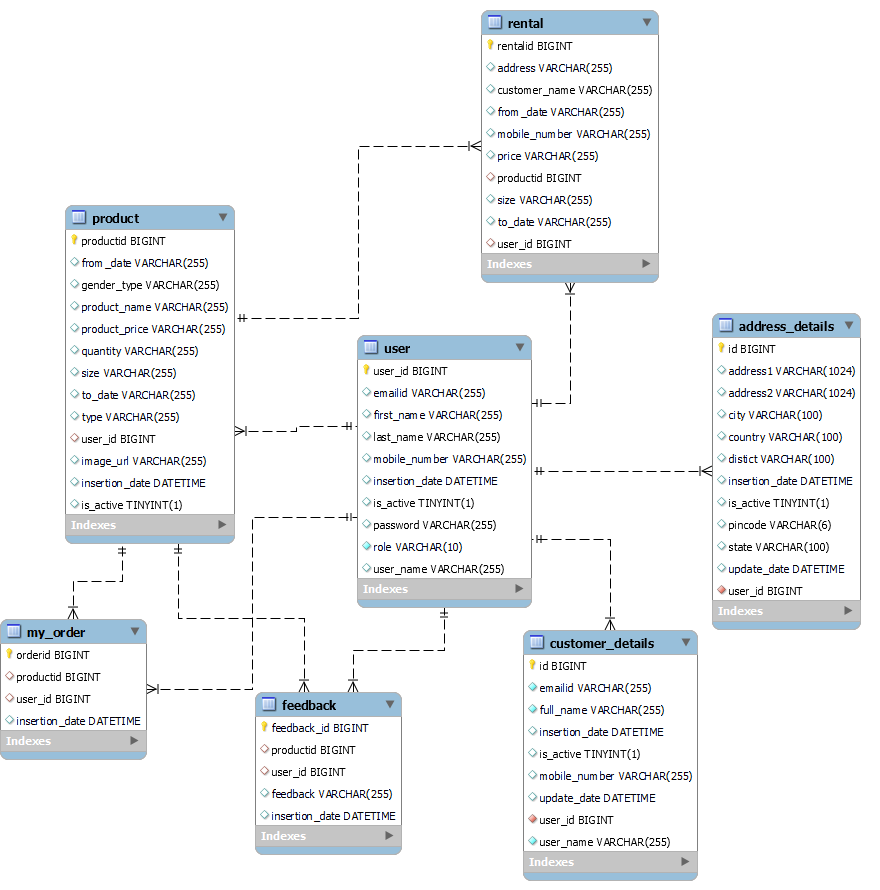


Figure 1: ER Diagram

1. **Table Structures:**
2. **Table name : user**

**Column name Type Null Key**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| user\_id | bigint | NO | PRI |  |  |
| emailid | varchar(255) | YES |  |  |  |
| first\_name | varchar(255) | YES |  |  |  |
| last\_name | varchar(255) | YES |  |  |  |
| mobile\_number | varchar(255) | YES |  |  |  |
| insertion\_date | datetime | YES |  |  |  |
| is\_active | tinyint(1) | YES |  | 1 |  |
| password | varchar(255) | YES |  |  |  |
| role | varchar(10) | NO |  |  |  |
| user\_name | varchar(255) | YES |  |  |  |

1. **Table name : customer\_details**

**Column name Type Null Key**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| id | bigint | NO | PRI |  |  |
| emailid | varchar(255) | NO |  |  |  |
| full\_name | varchar(255) | NO |  |  |  |
| insertion\_date | datetime | YES |  |  |  |
| is\_active | tinyint(1) | YES | 1 |  |  |
| mobile\_number | varchar(255) | YES |  |  |  |
| update\_date | datetime | YES |  |  |  |
| user\_id | bigint | NO | MUL |  |  |
| user\_name | varchar(255) | NO |  |  |  |

1. **Table name : product**

**Column name Type Null Key**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| gender\_type | varchar(255) | YES |  |  |  |
| product\_name | varchar(255) | YES |  |  |  |
| product\_price | varchar(255) | YES |  |  |  |
| quantity | varchar(255) | YES |  |  |  |
| size | varchar(255) | YES |  |  |  |
| to\_date | varchar(255) | YES |  |  |  |
| type | varchar(255) | YES |  |  |  |
| user\_id | bigint | YES | MUL |  |  |
| image\_url | varchar(255) | YES |  |  |  |
| insertion\_date | datetime | YES |  |  |  |
| is\_active | tinyint(1) | YES |  | 1 |  |

1. **Table name : rental**

**Column name Type Null Key**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| rentalid | bigint | NO | PRI |  |  |
| address | varchar(255) | YES |  |  |  |
| customer\_name | varchar(255) | YES |  |  |  |
| from\_date | varchar(255) | YES |  |  |  |
| mobile\_number | varchar(255) | YES |  |  |  |
| price | varchar(255) | YES |  |  |  |
| productid | bigint | YES | MUL |  |  |
| size | varchar(255) | YES |  |  |  |
| to\_date | varchar(255) | YES |  |  |  |
| user\_id | bigint | YES | MUL |  |  |

5. **Table name : my\_order**

**Column name Type Null Key**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| orderid | bigint | NO | PRI |  |  |
| productid | bigint | YES | MUL |  |  |
| user\_id | bigint | YES | MUL |  |  |
| insertion\_date | datetime | YES |  |  |  |

6. **Table name : address\_details**

**Column name Type Null Key**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| id | bigint | NO | PRI |  |  |
| address1 | varchar(1024) | YES |  |  |  |
| address2 | varchar(1024) | YES |  |  |  |
| city | varchar(100) | YES |  |  |  |
| country | varchar(100) | YES |  |  |  |
| distict | varchar(100) | YES |  |  |  |
| insertion\_date | datetime | YES |  |  |  |
| is\_active | tinyint(1) | YES |  | 1 |  |
| pincode | varchar(6) | YES |  |  |  |
| state | varchar(100) | YES |  |  |  |
| update\_date | datetime | YES |  |  |  |
| user\_id | bigint | NO | MUL |  |  |

7. **Table name : feedback**

**Column name Type Null Key**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| feedback\_id | bigint | NO | PRI |  |  |
| productid | bigint | YES | MUL |  |  |
| user\_id | bigint | YES | MUL |  |  |
| feedback | varchar(255) | YES |  |  |  |
| insertion\_date | datetime | YES |  |  |  |

1. **End to End Flow of Application:**

**Customer:**

* + Customer will **Login** to the portal or will have to **Register** if he is not a registered user.
  + After registration, Customer will login and Dashboard page will be displayed to him which will display the Product Category.
  + From that page Customer can click on the “**men, women and kids”** button as per his requirement.
  + After choosing category, Customer will select the required product from the list for rent and will enter the required details.
  + A “**Payment form”** will be displayed on the Website showing all the payment options.
  + On this dashboard, Customer will also be able to rent his clothes for that he will select "**own clothes rental**" page to add his clothes on website.
  + Customer can see his “**Order list**” and will be able to give “**feedback”**.

**Admin:**

* Admin will **Login** to the portal or will have to **Register** if he is not a registered.
* After login, Admin dashboard will be visible and he will be able to see the “**Order list**”.
* Admin will be also able to see “**customer list**” here.
* Admin can add product on website for rent.
* Admin can see the review of the products.

1. **Future Scope of Project:-**

* Verification of users.
* Improvement in design.
* Mobile Application.
* We can add more products to rent such as jewelleries and bags.
* Chat BOT .

**Thank You!**