BeautyProductStoreApplication

Presented by

[Sreedevi Palleni]
[EBEON1021447885]
[ER Vennila Ramesh]
[EBEON122151123]
[SV Eshwari]
[EBEON1221457628]

Trainer:

Indrakka Mali.Prof of
(Full Stack Developer)
EduBridge Learning Pvt Ltd

Contents

Introduction

The purpose of this project is to motivate both sellers and buyers to use this application for purchasing their beauty and skincare products. With exponential increase in business, it becomes a tedious task to maintain records of all products made available to different customers. Manual working of the system would not be beneficial for either the organization or the working individual. So, a database management system in the form of a API needs to be developed so as to perform all the manual tasks of beauty product store database through means of computers.

Problem Statement

- The objective is to develop a database management system such that:
- The system maintains details of all products such as name, ID, price.
- The system maintains details of stores provided with its name, address, ID and a brief description.
- The Restful CRUD API maintains details of both store and product details.
- The API's are used to create, retrieve, update and delete a store, and then tested using postman.
- The system keeps track of individual orders booked in the store.

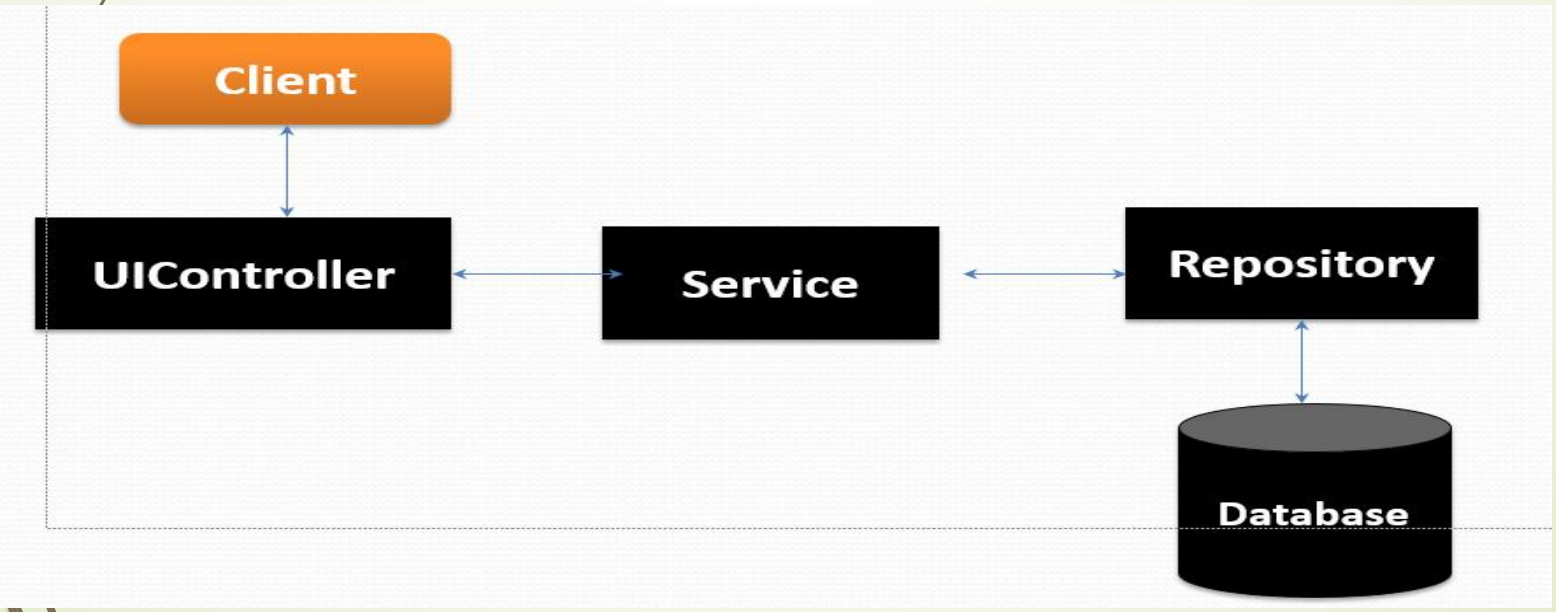
Backend Design

- Database design refers to the process of organization of data. The designer determines what data must be stored and how the data elements interrelate. With this information, they can begin to fit the data to the database accordingly.
- The four main types of databases are text databases, desktop database programs, relational database management systems (RDBMS) and NoSQL and object-oriented databases.

Code Design

❖ Client Side Processing

- Client side programming includes sort of interaction your website performs with the user via **browser**.
- Postman is an API client that makes it easy for developers to create, share, test and document APIs.
- This is done by allowing users to create and save simple and complex HTTP/s requests.
- **There are mainly three stages –**



❖ Workflow

•Client send a request to server for different options like inserting ,fetching ,update and delete a data.
•When client hit on request .Request goes to the Controller Class and use respective URL to take data .
•Then control of URL goes to the service Class And Conditions get Executed.
•For returning response to client use same sequence. Service To Control Class.
•Here Repository class is used to interact with database using Hibernate Technology.

❖ Association Using Hibernate:

- Hibernate understands the mappings that we add between objects and tables. It ensures that data is stored/retrieved from the database based on the mappings.

  Some of the Annotations Used in Project are:

  @Entity, @Id, @Generated Value, @Column

  @name,@length,@nullable etc…..

Exception Handling:

- Before you start designing your web page, you have to figure out what sort of a web page it's going to be and what contents it should have
- Some of the Annotations Used in Project are:
- @ControllerAdvice
- @ExceptionHandler
- @StoreNotFoundException
- @ProductNotFoundException. etc.,

Security

- In this project, Store Owner must smoothly navigate between the different pages he has access to.
- A new storing can be started upon providing the store Id number; if not registered there is a provision to directly go to the page for new store registration.
- Some of the Annotations Used in Project are:
- @EnableConfiguration
- @websecurity etc..,

Application Programming Interface(APIs)

APIS Under Director To Do :Product Entity

@GetMapping-"/admin/products/“

@PostMapping-"/admin/products/{storeId}“

@DeleteMapping-"/admin/products/{id}“

@PutMapping-"/admin/stores/{storeId}/products/{id}“

@GetMapping-"/customer/products/{id}“

@GetMapping-"/customer/products/price/{price}“

@GetMapping-"/customer/products/name/{name}"

APIS Under Director To Do :Store

@GetMapping("/stores/")

@PostMapping-"/stores/“

@DeleteMapping-"/stores/{id}“

@PutMapping-"/stores/{id}“

@GetMapping-"/customer/stores/{id}")

@GetMapping"/customer/stores/address/{address}“

@GetMapping-"/customer/stores/name/{name}

Software

- OS :Windows 10
- Tool: Spring tool suit(sts)

Technologies:

- Java : version 17
- Spring boot Framework: version 5.0
- My SQL
- Postman

Conclusion

- For a data and to store data in secure form.

THANK YOU