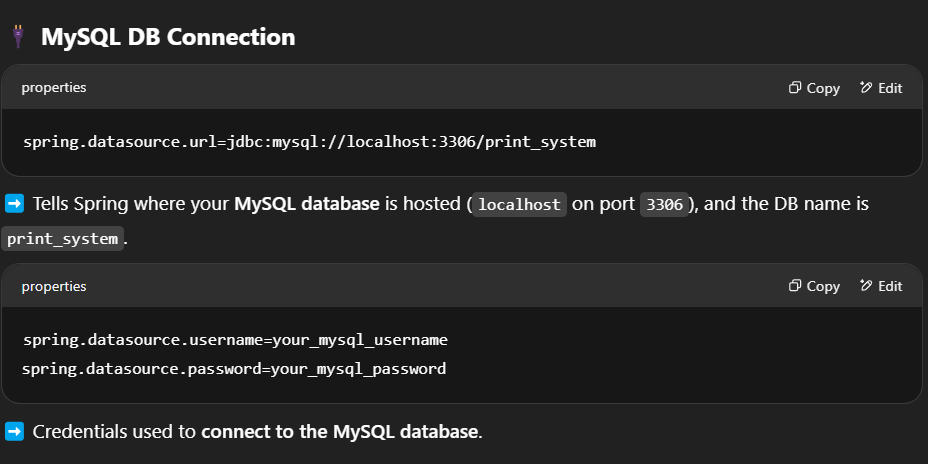
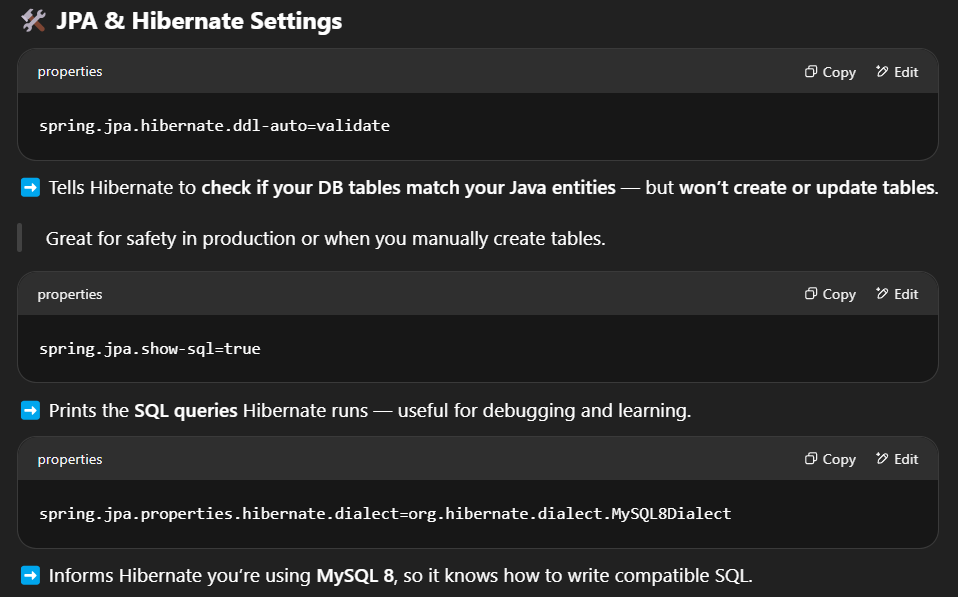
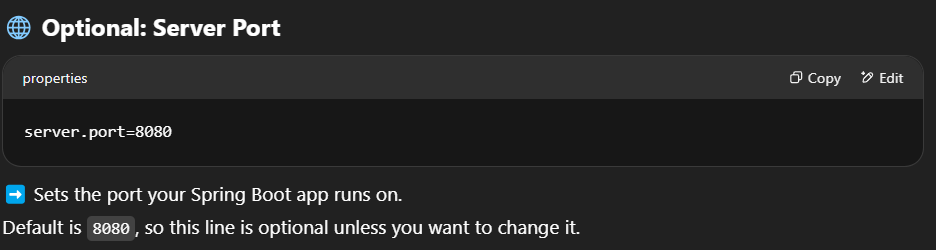
DataBase and Backend Connection:







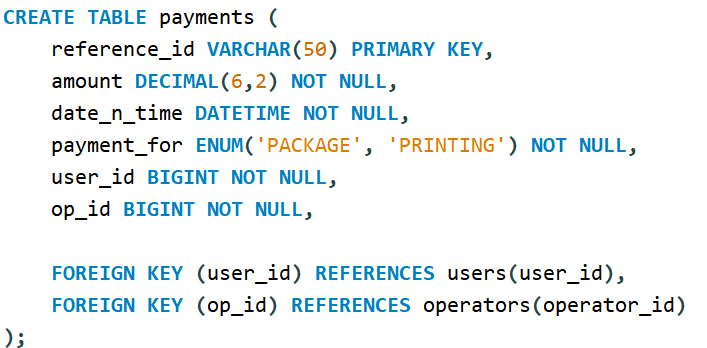
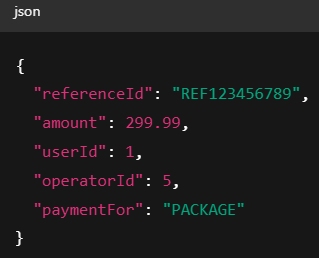
User Entity

Operator Registration

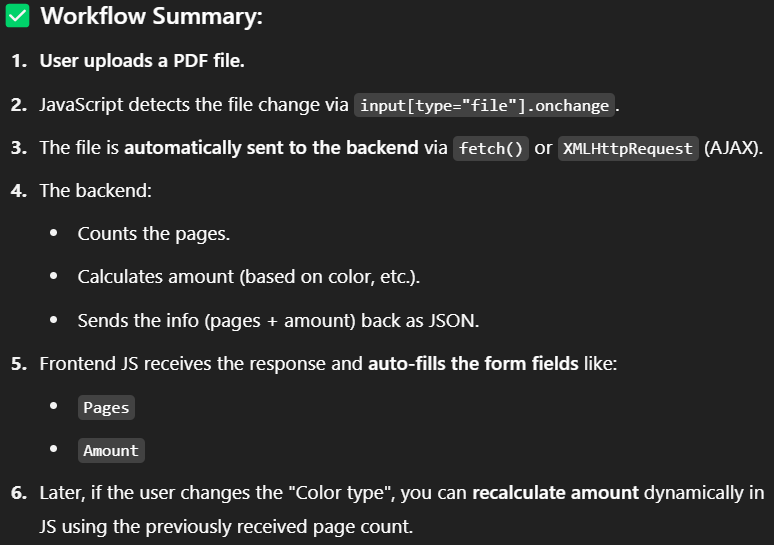
Operator Registration JSON should look like this.

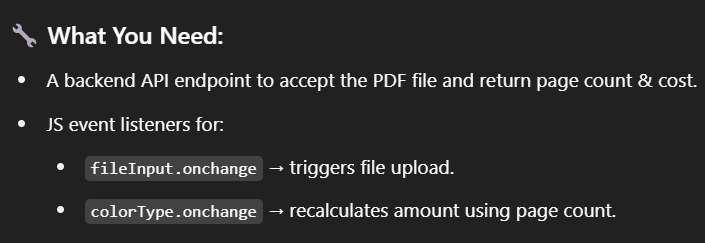
Payment

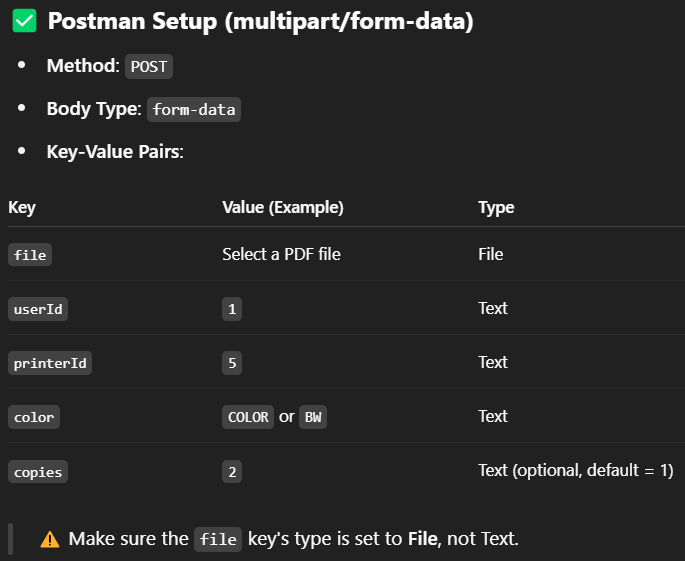
Add Status in the payment table to verify whether the payment is approved, rejected or verifying.

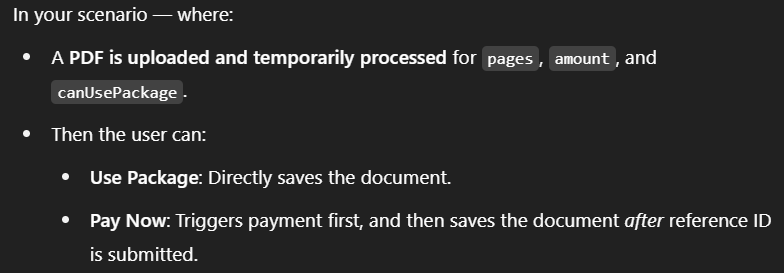
1. Create DTO for payment for package form. From the front-end Reference\_Id, user\_id, operator\_id, amount and paymentFor = Package will come and in the backend date\_n\_time and payment\_status will be generated.

PDF page count and amount (Max Size=10MB)

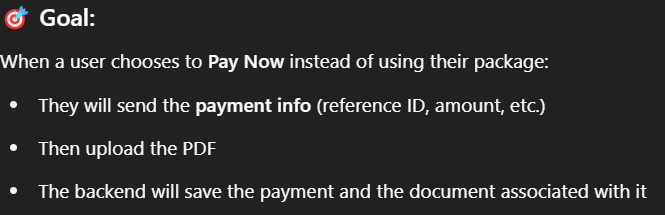


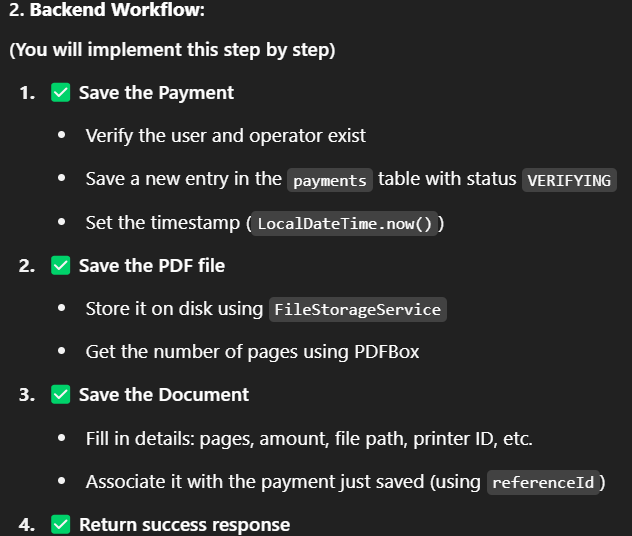


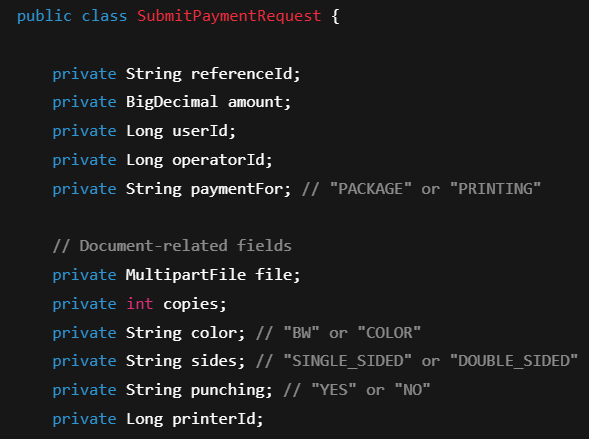


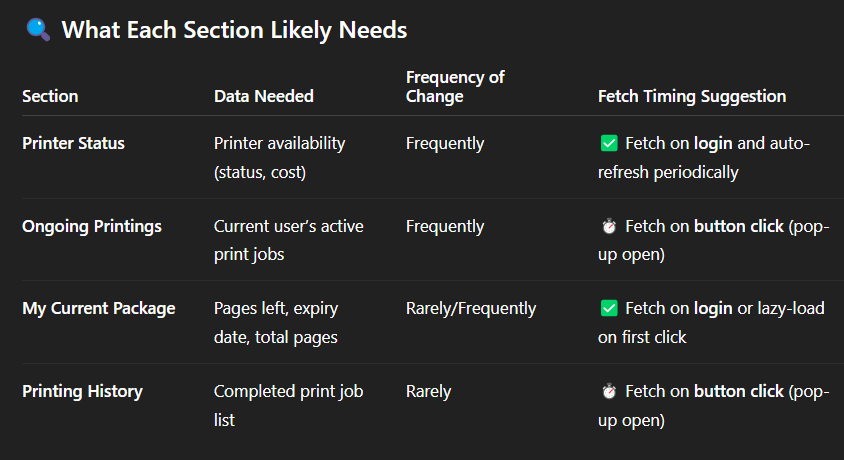
Use Package and Pay Now button

Use Package

Pay Now

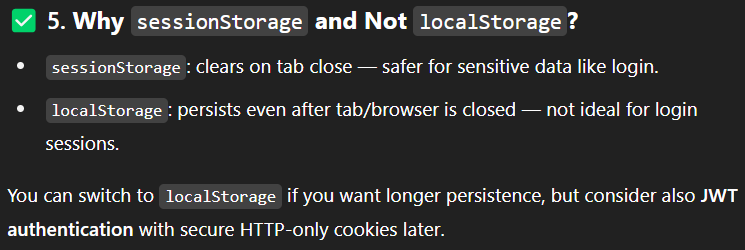


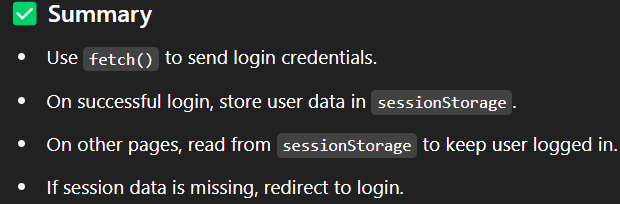


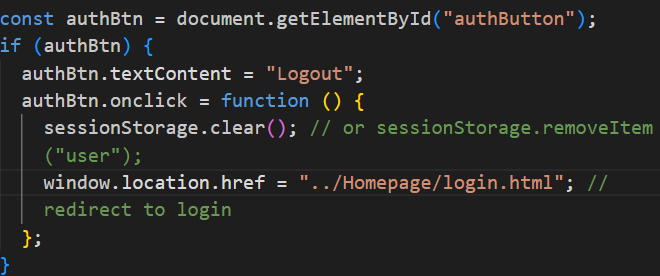
HomePage

On the Ongoing Printings Include Document and consider adding another field named Document Name in the document table.

In the Ongoing payments it will also show the amount even if the user used package. Fix it Later.

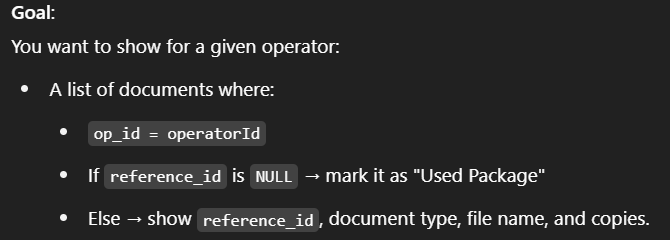
HomePage-Login Connection





This might be an issue if there are multiple users at a time. Not Sure though.

This is For Printer Op Dashboard



@Id  
@GeneratedValue(strategy = GenerationType.*IDENTITY*)  
@Column(name = "printer\_id")  
private Long printerId;  
  
@Column(name = "name", length = 30, nullable = false)  
private String name;  
  
@Column(name = "location", length = 100, nullable = false)  
private String location;  
  
@Enumerated(EnumType.*STRING*)  
@Column(name = "status", nullable = false)  
private PrinterStatus status = PrinterStatus.*NOT\_AVAILABLE*;  
  
@Column(name = "time\_per\_page\_bw", nullable = false)  
private double timePerPageBw;  
  
@Column(name = "time\_per\_page\_color", nullable = false)  
private double timePerPageColor;  
  
@Column(name = "cost\_bw", precision = 4, scale = 2, nullable = false)  
private BigDecimal costBw;  
  
@Column(name = "busy\_till")  
private LocalDateTime busyTill;  
  
@Column(name = "cost\_color", precision = 4, scale = 2, nullable = false)  
private BigDecimal costColor;  
  
@Enumerated(EnumType.*STRING*)  
@Column(name = "has\_package", nullable = false)  
private HasPackage hasPackage = HasPackage.*NO*;  
  
@Column(name = "package\_price", precision = 6, scale = 2)  
private BigDecimal packagePrice;  
  
@Column(name = "package\_page")  
private Integer packagePage;  
  
@Column(name="available\_till")  
private LocalDateTime availableTill;  
  
public LocalDateTime getAvailableTill() {  
 return availableTill;  
}  
  
public void setAvailableTill(LocalDateTime availableTill) {  
 this.availableTill = availableTill;  
}  
  
@OneToOne(fetch = FetchType.*LAZY*)  
@JoinColumn(name = "op\_id", nullable = false)  
private Operator operator;

This is what my printer entity looks like. Also the code you provided you didn’t separate the Controller and Service layer. So kindly separate then and give me one by one and one at a time.