**CODING**

**Sourcefolder—src/main/java**

**Package -----com.lms**

* **App.java**

package com.lms;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.boot.builder.SpringApplicationBuilder;

import org.springframework.context.ConfigurableApplicationContext;

import org.springframework.context.annotation.ComponentScan;

import org.springframework.context.annotation.PropertySource;

@SpringBootApplication

@ComponentScan("com.lms")

@PropertySource("classpath:/application.properties")

public class App {

public static void main(String[] args) {

ConfigurableApplicationContext context = new SpringApplicationBuilder(App.class).headless(false).run(args);

context.getBean(CallingService.class).call();

//SpringApplication.run(Test.class, args);

}

}

* **CallingService.java**

package com.lms;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Component;

import com.lms.entity.Book;

import com.lms.enums.BookTypeEnum;

import com.lms.enums.LoginTypeEnum;

import com.lms.service.BookService;

@Component

public class CallingService {

// @Autowired

// BookService bookService;

public void call() {

Book book = new Book(1, "Complete Java", "Kathy Sierra", 250, BookTypeEnum.AVAILABLE.name());

BookService service = new BookService();

String response = service.addBook(book, LoginTypeEnum.LIBRARIAN);

System.out.println(response);

}

}

**Package -----** **com.lms.common.service**

* **AbstractService.java**

package com.lms.common.service;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

import org.springframework.util.StringUtils;

import com.lms.entity.Login;

import com.lms.entity.Student;

import com.lms.entity.Teacher;

import com.lms.enums.LoginTypeEnum;

import com.lms.repository.LoginRepository;

import com.lms.repository.StudentRepository;

import com.lms.repository.TeacherRepository;

@Service

public abstract class AbstractService {

@Autowired

protected LoginRepository loginRepository;

@Autowired

protected StudentRepository studentRepository;

@Autowired

protected TeacherRepository teacherRepository;

public String addUserByAdmin(Login login) {

String response = "You dont have permission to add user, you are not ADMIN.";

if (login.getLoginType().equalsIgnoreCase(LoginTypeEnum.ADMIN.name()) && !checkUsername(login.getUsername())) {

Login login2 = loginRepository.save(login);

if (!StringUtils.isEmpty(login2)) {

response = "User successfully added";

}

} else {

response = "User already exist";

}

return response;

}

public String addStudent(Student student, LoginTypeEnum loginTypeEnum) {

String response = "You dont have permission to add user.";

if (!StringUtils.isEmpty(student) && !StringUtils.isEmpty(loginTypeEnum)) {

if (loginTypeEnum.equals(LoginTypeEnum.LIBRARIAN)

|| loginTypeEnum.equals(LoginTypeEnum.ADMIN) && (!checkUsername(student.getStudentUsername()))) {

Student stud = studentRepository.save(student);

if (!StringUtils.isEmpty(stud)) {

response = "Student added successfully...!!!";

}

}

}

return response;

}

public String addTeacher(Teacher teacher, LoginTypeEnum loginTypeEnum) {

String response = "You dont have permission to add user.";

if (!StringUtils.isEmpty(teacher) && !StringUtils.isEmpty(loginTypeEnum)) {

if (loginTypeEnum.equals(LoginTypeEnum.LIBRARIAN)

|| loginTypeEnum.equals(LoginTypeEnum.ADMIN) && (!checkUsername(teacher.getTeacherUsername()))) {

Teacher teacher1 = teacherRepository.save(teacher);

if (!StringUtils.isEmpty(teacher1)) {

response = "Teacher added successfully...!!!";

}

}

}

return response;

}

protected abstract boolean checkUsername(String username);

}

**Package -----** **com.lms.entity**

* **Book.java**

package com.lms.entity;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.Id;

import javax.persistence.Table;

@Entity

@Table(name = "BOOK")

public class Book {

@Id

@Column(name = "BOOK\_ID")

private Integer bookId;

@Column(name = "BOOK\_NAME", nullable = false)

private String bookName;

@Column(name = "BOOK\_AUTHOR", nullable = false)

private String bookAuthor;

@Column(name = "BOOK\_COST", nullable = false)

private double bookCost;

@Column(name = "STATUS", nullable = false)

private String status;

public Book() {

super();

}

public Book(Integer bookId, String bookName, String bookAuthor, double bookCost, String status) {

super();

this.bookId = bookId;

this.bookName = bookName;

this.bookAuthor = bookAuthor;

this.bookCost = bookCost;

this.status = status;

}

public Integer getBookId() {

return bookId;

}

public void setBookId(Integer bookId) {

this.bookId = bookId;

}

public String getBookName() {

return bookName;

}

public void setBookName(String bookName) {

this.bookName = bookName;

}

public String getBookAuthor() {

return bookAuthor;

}

public void setBookAuthor(String bookAuthor) {

this.bookAuthor = bookAuthor;

}

public double getBookCost() {

return bookCost;

}

public void setBookCost(double bookCost) {

this.bookCost = bookCost;

}

public String getStatus() {

return status;

}

public void setStatus(String status) {

this.status = status;

}

}

* **Inventory.java**

package com.lms.entity;

import java.util.Date;

import javax.persistence.CascadeType;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.JoinColumn;

import javax.persistence.OneToOne;

import javax.persistence.Table;

@Entity

@Table(name = "INVENTORY")

public class Inventory {

@Id

@Column(name = "INVENTORY\_ID")

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Integer inventoryId;

@Column(name = "BK\_ID", nullable = false)

private Integer bkId;

@Column(name = "BOOK\_COUNT", nullable = false)

private int bookCount;

@Column(name = "BOOK\_ADDED\_BY", nullable = false)

private String addedBy;

@Column(name = "BOOK\_ADDED\_DATE", nullable = false)

private Date addedDate;

@Column(name = "STATUS", nullable = false)

private String status;

@OneToOne(cascade = CascadeType.ALL)

@JoinColumn(name = "BK\_ID", referencedColumnName = "BOOK\_ID", insertable = false, updatable = false)

private Book book;

public Inventory() {

super();

}

public Inventory(Integer inventoryId, Integer bkId, Integer bookCount, String addedBy, Date addedDate,

String status) {

super();

this.inventoryId = inventoryId;

this.bkId = bkId;

this.bookCount = bookCount;

this.addedBy = addedBy;

this.addedDate = addedDate;

this.status = status;

}

public Inventory(Integer inventoryId, Integer bkId, Integer bookCount, String addedBy, Date addedDate,

String status, Book book) {

super();

this.inventoryId = inventoryId;

this.bkId = bkId;

this.bookCount = bookCount;

this.addedBy = addedBy;

this.addedDate = addedDate;

this.status = status;

this.book = book;

}

public Integer getInventoryId() {

return inventoryId;

}

public void setInventoryId(Integer inventoryId) {

this.inventoryId = inventoryId;

}

public Integer getBkId() {

return bkId;

}

public void setBkId(Integer bkId) {

this.bkId = bkId;

}

public String getAddedBy() {

return addedBy;

}

public void setAddedBy(String addedBy) {

this.addedBy = addedBy;

}

public int getBookCount() {

return bookCount;

}

public void setBookCount(int bookCount) {

this.bookCount = bookCount;

}

public Date getAddedDate() {

return addedDate;

}

public void setAddedDate(Date addedDate) {

this.addedDate = addedDate;

}

public String getStatus() {

return status;

}

public void setStatus(String status) {

this.status = status;

}

public Book getBook() {

return book;

}

public void setBook(Book book) {

this.book = book;

}

}

* **Issue.java**

package com.lms.entity;

import java.util.Date;

import javax.persistence.CascadeType;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.JoinColumn;

import javax.persistence.OneToOne;

import javax.persistence.Table;

import javax.persistence.Temporal;

import javax.persistence.TemporalType;

@Entity

@Table(name = "ISSUE")

public class Issue {

@Id

@Column(name = "ISSUE\_ID")

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Integer issueId;

@Column(name = "BK\_ID", nullable = false)

private Integer bookId;

@Column(name = "T\_ID")

private Integer teacherId;

@Column(name = "S\_ID")

private Integer studentId;

@Temporal(TemporalType.TIMESTAMP)

@Column(name = "ISSUE\_BOOK\_DATE", columnDefinition = "DATETIME", nullable = false)

private Date issueDate;

@Column(name = "RETURN\_BOOK\_DATE")

private Date returnDate;

@Column(name = "STATUS", nullable = false)

private String status;

@OneToOne(cascade = CascadeType.ALL)

@JoinColumn(name = "BK\_ID", referencedColumnName = "BOOK\_ID", insertable = false, updatable = false)

private Book book;

@OneToOne(cascade = CascadeType.ALL)

@JoinColumn(name = "S\_ID", referencedColumnName = "STUDENT\_ID", insertable = false, updatable = false)

private Student student;

@OneToOne(cascade = CascadeType.ALL)

@JoinColumn(name = "T\_ID", referencedColumnName = "TEACHER\_ID", insertable = false, updatable = false)

private Teacher teacher;

public Issue() {

super();

}

public Issue(Integer issueId, Integer bookId, Integer teacherId, Integer studentId, Date issueDate,

Date returnDate, String status, Book book, Student student, Teacher teacher) {

super();

this.issueId = issueId;

this.bookId = bookId;

this.teacherId = teacherId;

this.studentId = studentId;

this.issueDate = issueDate;

this.returnDate = returnDate;

this.status = status;

this.book = book;

this.student = student;

this.teacher = teacher;

}

public Integer getIssueId() {

return issueId;

}

public void setIssueId(Integer issueId) {

this.issueId = issueId;

}

public Integer getBookId() {

return bookId;

}

public void setBookId(Integer bookId) {

this.bookId = bookId;

}

public Integer getTeacherId() {

return teacherId;

}

public void setTeacherId(Integer teacherId) {

this.teacherId = teacherId;

}

public Integer getStudentId() {

return studentId;

}

public void setStudentId(Integer studentId) {

this.studentId = studentId;

}

public Date getIssueDate() {

return issueDate;

}

public void setIssueDate(Date issueDate) {

this.issueDate = issueDate;

}

public Date getReturnDate() {

return returnDate;

}

public void setReturnDate(Date returnDate) {

this.returnDate = returnDate;

}

public String getStatus() {

return status;

}

public void setStatus(String status) {

this.status = status;

}

public Book getBook() {

return book;

}

public void setBook(Book book) {

this.book = book;

}

public Student getStudent() {

return student;

}

public void setStudent(Student student) {

this.student = student;

}

public Teacher getTeacher() {

return teacher;

}

public void setTeacher(Teacher teacher) {

this.teacher = teacher;

}

}

* **Login.java**

package com.lms.entity;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.Table;

@Entity

@Table(name = "LOGIN")

public final class Login {

@Id

@Column(name = "LOGIN\_ID")

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Integer loginId;

@Column(name = "USERNAME", unique = true)

private String username;

@Column(name = "PASSWORD")

private String password;

@Column(name = "LOGIN\_TYPE")

private String loginType;

@Column(name = "STATUS")

private boolean status;

public Login() {

super();

}

public Login(String username, String password, String loginType, boolean status) {

super();

this.username = username;

this.password = password;

this.loginType = loginType;

this.status = status;

}

public Login(Integer loginId, String username, String password, String loginType, boolean status) {

super();

this.loginId = loginId;

this.username = username;

this.password = password;

this.loginType = loginType;

this.status = status;

}

public Integer getLoginId() {

return loginId;

}

public void setLoginId(Integer loginId) {

this.loginId = loginId;

}

public String getUsername() {

return username;

}

public void setUsername(String username) {

this.username = username;

}

public String getPassword() {

return password;

}

public void setPassword(String password) {

this.password = password;

}

public String getLoginType() {

return loginType;

}

public void setLoginType(String loginType) {

this.loginType = loginType;

}

public boolean isStatus() {

return status;

}

public void setStatus(boolean status) {

this.status = status;

}

@Override

public String toString() {

return "Login [loginId=" + loginId + ", username=" + username + ", password=" + password + ", loginType="

+ loginType + ", status=" + status + "]";

}

}

* **Student.java**

package com.lms.entity;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.Table;

@Entity

@Table(name = "STUDENT")

public class Student {

@Id

@Column(name = "STUDENT\_ID")

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Integer studentId;

@Column(name = "STUDENT\_USERNAME", nullable = false)

private String studentUsername;

@Column(name = "STUDENT\_NAME", nullable = false)

private String studentName;

@Column(name = "STUDENT\_ADDRESS", nullable = false)

private String studentAddress;

@Column(name = "STUDENT\_BRANCH", nullable = false)

private String branch;

@Column(name = "STATUS", nullable = false)

private boolean status;

public Student() {

super();

}

public Student(Integer studentId, String studentName, String studentAddress, String branch, boolean status,

String studentUsername) {

super();

this.studentId = studentId;

this.studentName = studentName;

this.studentAddress = studentAddress;

this.branch = branch;

this.status = status;

this.studentUsername = studentUsername;

}

public Integer getStudentId() {

return studentId;

}

public void setStudentId(Integer studentId) {

this.studentId = studentId;

}

public String getStudentUsername() {

return studentUsername;

}

public void setStudentUsername(String studentUsername) {

this.studentUsername = studentUsername;

}

public String getStudentName() {

return studentName;

}

public void setStudentName(String studentName) {

this.studentName = studentName;

}

public String getStudentAddress() {

return studentAddress;

}

public void setStudentAddress(String studentAddress) {

this.studentAddress = studentAddress;

}

public String getBranch() {

return branch;

}

public void setBranch(String branch) {

this.branch = branch;

}

public boolean isStatus() {

return status;

}

public void setStatus(boolean status) {

this.status = status;

}

}

* **Teacher.java**

package com.lms.entity;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.Table;

@Entity

@Table(name = "TEACHER")

public class Teacher {

@Id

@Column(name = "TEACHER\_ID", nullable = false)

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Integer teacherId;

@Column(name = "TEACHER\_USERNAME", nullable = false)

private String teacherUsername;

@Column(name = "TEACHER\_NAME", nullable = false)

private String teacherName;

@Column(name = "TEACHER\_ADDRESS", nullable = false)

private String teacherAddress;

@Column(name = "STATUS", nullable = false)

private boolean status;

public Teacher() {

super();

}

public Teacher(Integer teacherId, String teacherName, String teacherAddress, boolean status,

String teacherUsername) {

super();

this.teacherId = teacherId;

this.teacherName = teacherName;

this.teacherAddress = teacherAddress;

this.teacherUsername = teacherUsername;

this.status = status;

}

public Integer getTeacherId() {

return teacherId;

}

public void setTeacherId(Integer teacherId) {

this.teacherId = teacherId;

}

public String getTeacherUsername() {

return teacherUsername;

}

public void setTeacherUsername(String teacherUsername) {

this.teacherUsername = teacherUsername;

}

public String getTeacherName() {

return teacherName;

}

public void setTeacherName(String teacherName) {

this.teacherName = teacherName;

}

public String getTeacherAddress() {

return teacherAddress;

}

public void setTeacherAddress(String teacherAddress) {

this.teacherAddress = teacherAddress;

}

public boolean isStatus() {

return status;

}

public void setStatus(boolean status) {

this.status = status;

}

}

**Package ----- com.lms.enums**

* **BookTypeEnum.java**

package com.lms.enums;

public enum BookTypeEnum {

AVAILABLE, NOT\_AVAILABLE

}

* **InventoryTypeEnum.java**

package com.lms.enums;

public enum InventoryTypeEnum {

AVAILABLE, NOT\_AVAILABLE, OUT\_OF\_STOCK, COMING\_SOON

}

* **IssueTypeEnum.java**

package com.lms.enums;

public enum IssueTypeEnum {

ISSUED, RETURNED, RETURN\_PENDING

}

* **LoginTypeEnum.java**

package com.lms.enums;

public enum LoginTypeEnum {

ADMIN, STUDENT, TEACHER, LIBRARIAN, NOT\_AUTHORISED

}

**Package --- com.lms.repository**

* **BookRepository.java**

package com.lms.repository;

import org.springframework.data.repository.CrudRepository;

import com.lms.entity.Book;

public interface BookRepository extends CrudRepository<Book, Integer> {

public Book findByBookIdAndStatus(Integer bookId, String status);

public java.util.List<Book> findByBookNameAndStatus(String bookName, String status);

}

* **InventoryRepository.java**

package com.lms.repository;

import org.springframework.data.repository.CrudRepository;

import com.lms.entity.Inventory;

public interface InventoryRepository extends CrudRepository<Inventory, Integer> {

public Inventory findByInventoryId(Integer inventoryId);

public Inventory findByBkId(Integer bookId);

public java.util.List<Inventory> findByBookName(String bookName);

}

* **IssueRepository.java**

package com.lms.repository;

import org.springframework.data.repository.CrudRepository;

import com.lms.entity.Issue;

public interface IssueRepository extends CrudRepository<Issue, Integer> {

public java.util.List<Issue> findByStudentId(Integer studentId);

public java.util.List<Issue> findByTeacherId(Integer teacherId);

}

* **LoginRepository.java**

package com.lms.repository;

import org.springframework.data.repository.CrudRepository;

import com.lms.entity.Login;

public interface LoginRepository extends CrudRepository<Login, Integer> {

public Login findByUsername(String username);

public Login findByUsernameAndPasswordAndLoginTypeAndStatus(String username, String password, String loginType,

boolean status);

}

* **StudentRepository.java**

package com.lms.repository;

import org.springframework.data.repository.CrudRepository;

import com.lms.entity.Student;

public interface StudentRepository extends CrudRepository<Student, Integer> {

public Student findByStudentIdAndStudentAddress(Integer studentId, String address);

public Student findByStudentUsername(String username);

}

* **TeacherRepository.java**

package com.lms.repository;

import org.springframework.data.repository.CrudRepository;

import com.lms.entity.Teacher;

public interface TeacherRepository extends CrudRepository<Teacher, Integer> {

public Teacher findByTeacherUsername(String username);

}

**Package--- com.lms.service**

* **BookService.java**

package com.lms.service;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

import org.springframework.util.StringUtils;

import com.lms.entity.Book;

import com.lms.entity.Inventory;

import com.lms.enums.LoginTypeEnum;

import com.lms.repository.BookRepository;

import com.lms.repository.InventoryRepository;

@Service

public class BookService {

@Autowired

private BookRepository bookRepository;

@Autowired

private InventoryRepository inventoryRepository;

@Autowired

private InventoryService inventoryService;

public String addBook(Book book, LoginTypeEnum loginTypeEnum) {

String response = "You dont have permission to add user.";

if (!StringUtils.isEmpty(book) && !StringUtils.isEmpty(loginTypeEnum)) {

if (loginTypeEnum.equals(LoginTypeEnum.LIBRARIAN)

|| loginTypeEnum.equals(LoginTypeEnum.ADMIN) && (!isBookExist(book))) {

try {

Book books = bookRepository.save(book);

if (!StringUtils.isEmpty(books)) {

Inventory inv = inventoryService.addBookInventory(

inventoryRepository.findByBkId(books.getBookId()), LoginTypeEnum.LIBRARIAN.name());

if (!StringUtils.isEmpty(inv)) {

response = "Book added successfully...!!!";

} else {

response = "Unable to add book in inventory...";

}

} else {

response = "Book already exist...!!!";

}

} catch (Exception e) {

response = "Book already exist...!!!";

}

}

}

return response;

}

public Book updateBook(Book book) {

Book books = null;

if (!StringUtils.isEmpty(book)) {

books = new Book(book.getBookId(), book.getBookName(), book.getBookAuthor(), book.getBookCost(),

book.getStatus());

try {

Book bk = bookRepository.save(books);

if (!StringUtils.isEmpty(bk)) {

return bk;

}

} catch (Exception e) {

System.err.println("Inside catch block in Book");

return new Book();

}

}

return books;

}

private boolean isBookExist(Book book) {

boolean isExist = false;

if (!StringUtils.isEmpty(book) && !StringUtils.isEmpty(book.getBookId())) {

bookRepository.findOne(book.getBookId());

}

return isExist;

}

}

* **InventoryService.java**

package com.lms.service;

import java.util.Date;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

import org.springframework.util.StringUtils;

import com.lms.entity.Inventory;

import com.lms.entity.Issue;

import com.lms.enums.InventoryTypeEnum;

import com.lms.enums.LoginTypeEnum;

import com.lms.repository.InventoryRepository;

@Service

public class InventoryService {

@Autowired

private InventoryRepository inventoryRepository;

public Inventory addBookInventory(Inventory inv, String addedBy) {

Inventory inventory = null;

if (!StringUtils.isEmpty(inv) && !StringUtils.isEmpty(addedBy)) {

if (addedBy.equals(LoginTypeEnum.ADMIN) || addedBy.equals(LoginTypeEnum.LIBRARIAN)) {

Inventory inventorys = inventoryRepository.findByBkId(inv.getBkId());

if (!StringUtils.isEmpty(inventorys)) {

inventory = new Inventory(inventorys.getInventoryId(), inventorys.getBkId(),

inventorys.getBookCount(), addedBy, new Date(), InventoryTypeEnum.AVAILABLE.name());

return inventoryRepository.save(inventory);

} else {

return inventoryRepository.save(inv);

}

}

}

return inventory;

}

public Inventory bookCountOperation(Issue issue, boolean issued) {

Inventory tempInventory = null, inv = null;

if (!StringUtils.isEmpty(issue) && !StringUtils.isEmpty(issue.getBookId())) {

Inventory inventory = inventoryRepository.findByBkId(issue.getBookId());

tempInventory = new Inventory();

// If book got issued...

if (issued) {

if (inventory.getBookCount() == 1) {

tempInventory.setBookCount(0);

} else {

tempInventory.setBookCount(inventory.getBookCount() - 1);

}

// if book got returned...

} else {

if (inventory.getBookCount() == 0) {

tempInventory.setBookCount(1);

} else {

tempInventory.setBookCount(inventory.getBookCount() + 1);

}

}

try {

inv = inventoryRepository.save(tempInventory);

} catch (Exception e) {

inv = new Inventory();

}

} else {

inv = new Inventory();

}

return inv;

}

public boolean isBookAvalableInInventory(Issue issue) {

boolean isExist = false;

if (!StringUtils.isEmpty(issue) && !StringUtils.isEmpty(issue.getBookId())) {

Inventory inventory = inventoryRepository.findByBkId(issue.getBookId());

if (!StringUtils.isEmpty(inventory) && inventory.getBookCount() > 0) {

isExist = true;

}

}

return isExist;

}

}

* **IssueService.java**

package com.lms.service;

import java.util.Date;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

import org.springframework.util.StringUtils;

import com.lms.entity.Inventory;

import com.lms.entity.Issue;

import com.lms.enums.IssueTypeEnum;

import com.lms.enums.LoginTypeEnum;

import com.lms.repository.IssueRepository;

@Service

public class IssueService {

@Autowired

private IssueRepository issueRepository;

@Autowired

private InventoryService inventoryService;

public String issueBook(Issue issueBook) {

String response = null;

if (!StringUtils.isEmpty(issueBook) && inventoryService.isBookAvalableInInventory(issueBook)) {

issueBook.setStatus(IssueTypeEnum.ISSUED.name());

issueBook.setIssueDate(new Date());

Issue issue = issueRepository.save(issueBook);

if (!StringUtils.isEmpty(issue)) {

Inventory inventory = inventoryService.bookCountOperation(issue, true);

if (!StringUtils.isEmpty(inventory)) {

response = "Book has been issued and your issue id is " + issue.getIssueId();

} else {

response = "Not able to issue book ...";

}

}

} else {

response = "Book not available";

}

return response;

}

public String returnBook(Issue issueBook, String returnBy) {

Issue returnBook;

String response = null;

if (!StringUtils.isEmpty(issueBook)) {

returnBook = new Issue();

returnBook.setIssueId(issueBook.getIssueId());

returnBook.setStudentId(issueBook.getStudentId());

returnBook.setTeacherId(issueBook.getTeacherId());

returnBook.setIssueDate(new Date());

if (returnBy.equals(LoginTypeEnum.LIBRARIAN.name()) || returnBy.equals(LoginTypeEnum.ADMIN.name())) {

returnBook.setStatus(IssueTypeEnum.RETURNED.name());

} else {

returnBook.setStatus(IssueTypeEnum.RETURN\_PENDING.name());

}

Issue returnBk = issueRepository.save(returnBook);

if (!StringUtils.isEmpty(returnBk)) {

Inventory inventory = inventoryService.bookCountOperation(returnBk, false);

if (!StringUtils.isEmpty(inventory)) {

response = "Book has been issued and your issue id is " + returnBk.getIssueId();

} else {

response = "Not able to issue book ...";

}

}

} else {

response = "Book not available";

}

return response;

}

}

* **LoginService.java**

package com.lms.service;

import org.springframework.stereotype.Service;

import org.springframework.util.StringUtils;

import com.lms.common.service.AbstractService;

import com.lms.entity.Login;

import com.lms.enums.LoginTypeEnum;

@Service

public class LoginService extends AbstractService {

public String loginAs(Login login) {

String response = LoginTypeEnum.NOT\_AUTHORISED.name();

Login loginObj = loginCheck(login);

if (!StringUtils.isEmpty(loginObj) && !StringUtils.isEmpty(loginObj.getLoginType())) {

response = loginObj.getLoginType();

}

return response;

}

private Login loginCheck(Login login) {

Login loginObj = null;

if (!StringUtils.isEmpty(login)) {

loginObj = loginRepository.findByUsernameAndPasswordAndLoginTypeAndStatus(login.getUsername(),

login.getPassword(), login.getLoginType(), login.isStatus());

if (!StringUtils.isEmpty(loginObj)) {

loginObj.setPassword(null);

System.out.println("Successfully logged in...");

}

}

return loginObj;

}

@Override

protected boolean checkUsername(String username) {

boolean isUsernameExist = false;

if (!StringUtils.isEmpty(username)) {

Login login2 = loginRepository.findByUsername(username);

if (!StringUtils.isEmpty(login2) && !StringUtils.isEmpty(login2.getUsername())) {

isUsernameExist = true;

}

}

return isUsernameExist;

}

}

* **StudentService.java**

package com.lms.service;

import org.springframework.stereotype.Service;

import org.springframework.util.StringUtils;

import com.lms.common.service.AbstractService;

import com.lms.entity.Student;

@Service

public class StudentService extends AbstractService {

@Override

protected boolean checkUsername(String username) {

boolean isUsernameExist = false;

if (!StringUtils.isEmpty(username)) {

Student student = studentRepository.findByStudentUsername(username);

if (!StringUtils.isEmpty(student) && !StringUtils.isEmpty(student.getStudentUsername())) {

isUsernameExist = true;

}

}

return isUsernameExist;

}

public String updateStudent(Student student) {

Student studentTemp = null;

String response = null;

if (!StringUtils.isEmpty(student) && !StringUtils.isEmpty(student.getStudentUsername())) {

studentTemp = new Student(student.getStudentId(), student.getStudentName(), student.getStudentAddress(),student.getBranch(), student.isStatus(), student.getStudentUsername());

try {

Student stud = studentRepository.save(studentTemp);

if (!StringUtils.isEmpty(stud)) {

response = "Student details successfully updated...";

} else {

response = "Unable to update it...please provide values.";

}

} catch (Exception e) {

System.err.println("Inside Student Catch...");

response = "Unable to update it...please provide values.";

}

} else {

response = "Please provide valid username...!!!";

}

return response;

}

}

* **TeacherService.java**

package com.lms.service;

import org.springframework.stereotype.Service;

import org.springframework.util.StringUtils;

import com.lms.common.service.AbstractService;

import com.lms.entity.Teacher;

@Service

public class TeacherService extends AbstractService {

@Override

protected boolean checkUsername(String username) {

boolean isUsernameExist = false;

if (!StringUtils.isEmpty(username)) {

Teacher student = teacherRepository.findByTeacherUsername(username);

if (!StringUtils.isEmpty(student) && !StringUtils.isEmpty(student.getTeacherUsername())) {

isUsernameExist = true;

}

}

return isUsernameExist;

}

public String updateTeacher(Teacher teacher) {

Teacher teacherTemp = null;

String response = null;

if (!StringUtils.isEmpty(teacher) && !StringUtils.isEmpty(teacher.getTeacherUsername())) {

teacherTemp = new Teacher(teacher.getTeacherId(), teacher.getTeacherName(), teacher.getTeacherAddress(),teacher.isStatus(), teacher.getTeacherUsername());

try {

Teacher stud = teacherRepository.save(teacherTemp);

if (!StringUtils.isEmpty(stud)) {

response = "Teacher details successfully updated...";

} else {

response = "Unable to update it...please provide values.";

}

} catch (Exception e) {

System.err.println("Inside Teacher Catch...");

response = "Unable to update it...please provide values.";

}

} else {

response = "Please provide valid username...!!!";

}

return response;

}

}