**SOURCE CODE**

User login.html:-

<body>

<div class="container">

<div class="row">

<div class="col-sm-9 col-md-7 col-lg-5 mx-auto">

<div class="card border-0 shadow rounded-3 my-5">

<div class="card-body p-4 p-sm-5">

<h5 class="card-title text-center mb-5 fw-light fs-5">Sign In</h5>

<form>

<div class="form-floating mb-3">

<input [(ngModel)]="loginUserId" id="loginUserId" type="text" name="loginUserId" placeholder="User Login Id"

class="form-control bg-white border-left-0 border-md" required>

<label for="floatingInput">User ID</label>

</div>

<div class="form-floating mb-3">

<input [(ngModel)]="loginUserPassword" id="loginUserPassword" type="password" name="loginUserPassword" placeholder="Login Password"

class="form-control bg-white border-left-0 border-md" required>

<label for="floatingPassword">Password</label>

</div>

<div class="d-grid">

<button (click)="loginUser()" class="btn btn-primary btn-login text-uppercase fw-bold" type="submit">Sign

in</button>

</div>

<hr class="my-4">

<div class="d-grid mb-2">

<p class="text-muted font-weight-bold">Not a Registered User?</p><br>

<button class="btn btn-google btn-login text-uppercase fw-bold" routerLink="/registration">

<i class="fab fa-google me-2"></i> Click here to sign up

</button><br>

<a routerLink="admin">Administrator Login</a>

<a routerLink="user-account-create">Click here to create account</a>

</div>

</form>

</div>

</div>

</div>

</div>

</div>

</body>

User profile.html:-

<head>

<link rel="stylesheet" type="text/css" href="https://maxcdn.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css" />

</head>

<body>

<div class="card col-6 offset-3" style="width: 25rem;">

<img class="card-img-top" src="https://www.shutterstock.com/image-vector/setting-icon-600w-639144637.jpg" alt="Card image" width="100" height="300">

<div class="card-header">My Profile</div>

<div class="card-body">

<h4 class="card-title">Settings</h4>

<p class="card-text">

<small class="col-lg-12">Login User Name</small>

<input readonly [(ngModel)]="userAccountHolderName" id="userAccountHolderName" type="text"

name="userAccountHolderName" placeholder="User Login Id"

class="form-control bg-white border-left-0 border-md" required>

<small class="col-lg-12">Login User Account Number</small>

<input readonly [(ngModel)]="userAccountNumber" id="userAccountNumber" type="text"

name="userAccountNumber" placeholder="Account Number"

class="form-control bg-white border-left-0 border-md" required>

<small class="col-lg-12">Login User Id</small>

<input readonly [(ngModel)]="loginUserId" id="loginUserId" type="text" name="loginUserId"

placeholder="User Login Id" class="form-control bg-white border-left-0 border-md" required>

<small class="col-lg-12">Update Login User Password</small>

<input [(ngModel)]="loginUserPassword" id="loginUserPassword"

name="loginUserPassword" placeholder="Login Password"

class="form-control bg-white border-left-0 border-md myPadding" required><br>

<div>

<a class="btn btn-primary btn-block py-2" (click)="updateLoginPassword()">

<span class="font-weight-bold text-white">Update</span>

</a>

</div>

</div>

</div>

</body>

User registration.html :-

<div class="container myDiv">

<div class="row py-5 align-items-center">

<div class="col-md-5 pr-lg-5 mb-5 mb-md-0 text-center">

<h1 class="display-3">ICIN Bank</h1>

</div>

<div class="col-md-7 col-lg-6 ml-auto">

<form>

<div class="row">

<div class="form">

<h1 class="display-4">Register</h1>

</div>

<div class="form-floating mb-3">

<input [(ngModel)]="firstName" id="firstName" type="text" name="firstname" placeholder="First Name" class="form-control bg-white border-left-0 border-md" required>

<label for="firstname">First Name</label>

</div>

<div class="form-floating mb-3">

<input [(ngModel)]="lastName" id="lastName" type="text" name="lastname" placeholder="Last Name"

class="form-control bg-white border-left-0 border-md" required>

<label for="lastname">Last Name</label>

</div>

<div class="form-floating mb-3">

<input [(ngModel)]="accountNumber" id="accountNumber" type="text" name="accountNumber" placeholder="Account Number"

class="form-control bg-white border-left-0 border-md" required>

<label for="accountNumber">Enter the 9 digit account number provided by your branch representative</label>

</div>

<div class="form-group col-lg-12 mb-0" (click)="generateID()">

<a class="btn btn-success btn-block py-2">

<span class="font-weight-bold text-white">Click to obtain your new user ID</span><br>

</a>

</div>

<div class="input-group col-lg-12 mb-4">

<input value="{{ loginUserID }}" id="loginUserId" type="text" name="loginUserId" placeholder="Login User ID" readonly

class="form-control bg-white border-left-0 border-md">

</div>

<div class="input-group col-lg-6 mb-4">

<input [(ngModel)]="loginUserPassword1" id="password" type="password" name="password" placeholder="Enter a password of your choice"

class="form-control bg-white border-left-0 border-md" required>

</div>

<div class="input-group col-lg-6 mb-4">

<input [(ngModel)]="loginUserPassword2" id="passwordConfirmation" type="password" name="passwordConfirmation"

placeholder="Re-enter the password" class="form-control bg-white border-left-0 border-md" required>

</div>

<div class="form-group col-lg-12 mx-auto mb-0" (click)="register()">

<a class="btn btn-primary btn-block py-2">

<span class="font-weight-bold text-white">Register Your Account For Net Banking</span>

</a>

</div>

<div class="form-group col-lg-12 mx-auto d-flex align-items-center my-4">

<div class="border-bottom w-100 ml-5"></div>

<span class="px-2 small text-muted font-weight-bold text-muted">OR</span>

<div class="border-bottom w-100 mr-5"></div>

</div>

<div class="text-center w-100">

<p class="text-muted font-weight-bold">Already Registered? <a routerLink=""

class="text-primary ml-2">Login</a></p>

</div>

</div>

</form>

</div>

</div>

</div>

User transaction .html :-

<div class="col-md-12 col-lg-12 ml-auto">

<table class="table table-hover" id="passbook">

<thead class="thead-dark">

<tr>

<th scope="col" class="border-right">Serial #</th>

<th scope="col">Date</th>

<th scope="col">Details</th>

<th scope="col">Transaction Amount</th>

<th scope="col">Transaction Status</th>

</tr>

</thead>

<tbody>

<tr \*ngFor="let transaction of toShowTransactions">

<td scope="row" class="border-right">{{transaction['id']}}</td>

<td>{{transaction['transferDate']}}</td>

<td>{{generateRemarks(

transaction['fromAccountNumber'],

transaction['toAccountNumber'],

transaction['fromAccountHolderName'],

transaction['toAccountHolderName'],

transaction['fromAccountType'],

transaction['toAccountType'],

transaction['transferMessage']

)

}}</td>

<td class="{{amountColor}} font-weight-bold"> {{generateTransferAmount(

transaction['transferAmount'],

transaction['fromAccountNumber'],

transaction['toAccountNumber']

)}}</td>

<td>{{generateStatus(transaction['transferStatus'])}}</td>

</tr>

</tbody>

</table>

</div>

<ng-template #content let-modal>

<div class="modal-header">

<h4 class="modal-title" id="modal-basic-title">Filter Transactions</h4>

<button type="button" class="close" aria-label="Close" (click)="modal.dismiss('Cross click')">

<span aria-hidden="true">&times;</span>

</button>

</div>

<div class="modal-body">

<form>

<div class="form-group">

<label for="dateOfBirth">Start Date</label>

<div class="input-group">

<input [(ngModel)]="startDate" id="startDate" type="date" class="form-control" placeholder="dd/mm/yyyy" name="startDate">

</div>

<br>

<label for="dateOfBirth">End Date</label>

<div class="input-group">

<input [(ngModel)]="endDate" id="endDate" type="date" class="form-control" placeholder="dd/mm/yyyy" name="endDate">

</div>

</div>

</form>

</div>

<div class="modal-footer">

<button type="button" class="btn btn-outline-dark" (click)="modal.close('Clear Filter')">Clear Filter</button>

<button type="button" class="btn btn-outline-dark" (click)="modal.close('Filter')">Filter</button>

</div>

</ng-template>

Admin login.html :-

<body>

<div class="container">

<div class="row">

<div class="col-sm-9 col-md-7 col-lg-5 mx-auto">

<div class="card border-0 shadow rounded-3 my-5">

<div class="card-body p-4 p-sm-5">

<h5 class="card-title text-center mb-5 fw-light fs-5"> Administrator Portal</h5>

<form>

<div class="form-floating mb-3">

<input [(ngModel)]="adminUsername" id="adminUsername" type="text" name="adminUsername" placeholder="Admin User Name"

class="form-control bg-white border-left-0 border-md" required>

<label for="floatingInput">User ID</label>

</div>

<div class="form-floating mb-3">

<input [(ngModel)]="adminPassword" id="adminPassword" type="password" name="adminPassword" placeholder="Admin Password"

class="form-control bg-white border-left-0 border-md" required>

<label for="floatingPassword">Password</label>

</div>

<div class="d-grid">

<a (click)="loginAdmin()" class="btn btn-primary btn-block py-2">

<span class="font-weight-bold text-white">Login</span>

</a>

</div>

</form>

</div>

</div>

</div>

</div>

</div>

</body>

Admin account manage.html :-

<div class="container">

<div class="card">

<table class="table" id="unblock">

<tr class="table-active">

<th scope="col" class="border-right">Serial #</th>

<th scope="col">Account Number</th>

<th scope="col">Account Holder Name</th>

<th scope="col">Operation</th>

</tr>

<tr \*ngFor="let user of toShowUsers">

<td scope="row" class="border-right">{{user['accountLoginUserId']}}</td>

<td>{{user['accountNumber']}}</td>

<td>{{user['accountHolderName']}}</td>

<td>

<button (click)="unblock(user['accountLoginUserId'])" class="bg-danger btn btn-danger">Unblock user</button>

</td>

</tr>

</table>

</div>

</div>

Admin dashboard.html :-

<body>

<nav class="navbar navbar-expand-lg navbar-light" style="background-color: #4ba0dd;">

<span class="navbar-brand">Admin Portal</span>

<ul class="navbar-nav ms-auto">

<li class="nav-item">

<button class="btn btn-danger" (click)="onLogout()">Logout</button>

</li>

</ul>

</nav><br><br><br>

<div class="container">

<div class="card" style="width: 25rem;">

<img class="card-img top" src="https://plus.unsplash.com/premium\_photo-1671306963320-c368eacbd44f?ixlib=rb-4.0.3&ixid=MnwxMjA3fDB8MHxzZWFyY2h8MXx8YmFua3xlbnwwfHwwfHw%3D&auto=format&fit=crop&w=500&q=60">

<div class="card-body">

<div class="btn btn-primary btn-login" routerLink="/admin-cheque">Approve cheque book requests</div><br><br>

<div class="btn btn-primary btn-login" routerLink="/admin-transact">Approve transactions</div><br><br>

<div class="btn btn-primary btn-login" routerLink="/admin-manage-accounts">Manage blocked accounts</div><br><br>

<div class="btn btn-primary btn-login" routerLink="/admin-netbank-access">Enable net banking</div><br><br>

<div class="btn btn-primary btn-login" routerLink="/admin-threat">Block threat accounts</div><br><br>

</div>

</div>

</div>

</body>

Account service.java

package com.icinbank.service;

import java.util.List;

import com.icinbank.model.Accounts;

public interface AccountsService {

public List<Accounts> getAllAccounts();

public Accounts getAccount(String accountNumber);

public boolean isNetBankingReg(String accountNumber);

public String putAccount(Accounts account);

public String updateAccountNetBankingStatus(int status, String accountNumber);

public List<Accounts> getAllUnregisteredUsers();

public String depositPrimary(int balance, String accountNumber);

public String depositSavings(int balance, String accountNumber);

public String withdrawPrimary(int balance, String accountNumber);

public String withdrawSavings(int balance, String accountNumber);

}

Account controller.java

package com.icinbank.controller;

import java.util.List;

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

import org.springframework.web.bind.annotation.CrossOrigin;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.PathVariable;

import org.springframework.web.bind.annotation.PostMapping;

import org.springframework.web.bind.annotation.RequestBody;

import org.springframework.web.bind.annotation.RestController;

import com.icinbank.model.Accounts;

import com.icinbank.model.Message;

import com.icinbank.service.AccountsService;

@CrossOrigin(origins = "http://localhost:4200")

@RestController

public class AccountsController {

@Autowired

private AccountsService accountsService;

@GetMapping("/accounts")

public List<Accounts> getAllAccounts() {

return this.accountsService.getAllAccounts();

}

@GetMapping("/accounts/{accountNumber}")

public Accounts getAccount(@PathVariable String accountNumber) {

return this.accountsService.getAccount(accountNumber);

}

@GetMapping("/accounts/isNetBankingActivated/{accountNumber}")

public boolean checkIfNetBankingActivated(@PathVariable String accountNumber) {

return this.accountsService.isNetBankingReg(accountNumber);

}

@PostMapping("/accounts/add-account")

public Message addAccount(@RequestBody Accounts account) {

this.accountsService.putAccount(account);

return new Message("Account Added!");

}

@GetMapping("/accounts/netbankingenable/{status}/{accountNumber}")

public Message enableNetBanking(@PathVariable int status, @PathVariable String accountNumber) {

Message message = new Message(this.accountsService.updateAccountNetBankingStatus(1, accountNumber));

return message;

}

@GetMapping("/accounts/unregisteredUsers")

public List<Accounts> getUnregisteredAccounts() {

return this.accountsService.getAllUnregisteredUsers();

}

@GetMapping("/accounts/depositSavings/{balance}/{accountNumber}")

public Message depositSavings(@PathVariable int balance, @PathVariable String accountNumber) {

Message message = new Message(this.accountsService.depositSavings(balance, accountNumber));

return message;

}

@GetMapping("/accounts/depositPrimary/{balance}/{accountNumber}")

public Message depositPrimary(@PathVariable int balance, @PathVariable String accountNumber) {

Message message = new Message(this.accountsService.depositPrimary(balance, accountNumber));

return message;

}

@GetMapping("/accounts/withdrawSavings/{balance}/{accountNumber}")

public Message withdrawSavings(@PathVariable int balance, @PathVariable String accountNumber) {

Message message = new Message(this.accountsService.withdrawSavings(balance, accountNumber));

return message;

}

@GetMapping("/accounts/withdrawPrimary/{balance}/{accountNumber}")

public Message withdrawPrimary(@PathVariable int balance, @PathVariable String accountNumber) {

Message message = new Message(this.accountsService.withdrawPrimary(balance, accountNumber));

return message;

}

}

Transcation controller.java

package com.icinbank.controller;

import java.util.List;

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

import org.springframework.web.bind.annotation.CrossOrigin;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.PathVariable;

import org.springframework.web.bind.annotation.PostMapping;

import org.springframework.web.bind.annotation.RequestBody;

import org.springframework.web.bind.annotation.RestController;

import com.icinbank.model.FrontendTransaction;

import com.icinbank.model.Message;

import com.icinbank.model.Transactions;

import com.icinbank.service.TransactionsService;

@CrossOrigin(origins = "http://localhost:4200")

@RestController

public class TransactionController {

@Autowired

private TransactionsService transactionsService;

@PostMapping("/transactions/putTransaction")

public Message putTransaction(@RequestBody Transactions transactions) {

Message message = new Message(this.transactionsService.addTransaction(transactions));

return message;

}

@GetMapping("/transactions/{accountNumber}")

public List<FrontendTransaction> getTransactionsByAccountNumber(@PathVariable String accountNumber) {

return this.transactionsService.getTransactionsForAccountNumber(accountNumber);

}

@GetMapping("/admin/get-all-pending-transactions")

public List<Transactions> getPendingTransactions() {

return this.transactionsService.getAllPendingTransactions();

}

@GetMapping("/admin/allow/transaction/{id}")

public Message permitTransaction(@PathVariable int id) {

Message message = new Message(this.transactionsService.updateTransaction(id));

return message;

}

@GetMapping("/transactions/{accountNumber}/{startDate}/{endDate}")

public List<FrontendTransaction> getFilteredTransactions(

@PathVariable String accountNumber,

@PathVariable String startDate,

@PathVariable String endDate

) {

return this.transactionsService.getFilteredTransactions(accountNumber, startDate, endDate);

}

}

User controller.java

package com.icinbank.controller;

import java.util.List;

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

import org.springframework.web.bind.annotation.CrossOrigin;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.PathVariable;

import org.springframework.web.bind.annotation.PostMapping;

import org.springframework.web.bind.annotation.RequestBody;

import org.springframework.web.bind.annotation.RestController;

import com.icinbank.model.Accounts;

import com.icinbank.model.Cheque;

import com.icinbank.model.Message;

import com.icinbank.model.Users;

import com.icinbank.service.ChequeService;

import com.icinbank.service.UsersService;

@CrossOrigin(origins = "http://localhost:4200")

@RestController

public class UsersController {

@Autowired

private UsersService usersService;

@Autowired

private ChequeService chequeService;

@GetMapping("/users")

public List<Users> getAllAccounts() {

return this.usersService.getAllUser();

}

@GetMapping("/users/{loginUserId}")

public Users getUserViaUserId(@PathVariable String loginUserId) {

return this.usersService.getUserViaUserId(loginUserId);

}

@GetMapping("/users/check/{loginUserId}/{loginUserPassword}")

public Message checkUserByIdAndPassword(@PathVariable String loginUserId, @PathVariable String loginUserPassword) {

Message message = new Message(this.usersService.checkUserByIdAndPassword(loginUserId, loginUserPassword));

return message;

}

@PostMapping("/users/addUser")

public Message putUser(@RequestBody Users user) {

Message message = new Message(this.usersService.putUser(user));

return message;

}

@GetMapping("/users/update/loginPassword/{newPassword}/{accountNumber}")

public Message updateLoginPassword(@PathVariable String newPassword, @PathVariable String accountNumber) {

Message message = new Message(this.usersService.updateLoginPassword(newPassword, accountNumber));

return message;

}

@GetMapping("/users/request-cheque-book/{accountNumber}/{accountType}")

public Message requestChequeBook(@PathVariable String accountNumber, @PathVariable String accountType) {

Message message = new Message(this.chequeService.generateChequeBookRequest(accountNumber, accountType));

return message;

}

@GetMapping("/users/cheque-books/{accountNumber}")

public List<Cheque> getChequeBooksIssuedByAccountNumber(@PathVariable String accountNumber) {

return this.chequeService.getAllChequeBookIssued(accountNumber);

}

@GetMapping("/admin/cheque-books-requests")

public List<Cheque> getAllChequeBookRequests() {

return this.chequeService.getAllPendingChequeBookRequests();

}

@GetMapping("/admin/cheque-books-requests/accept/{chequeBookNumber}")

public Message acceptChequeBookRequest(@PathVariable String chequeBookNumber) {

Message message = new Message(this.chequeService.changeChequeBookStatus(chequeBookNumber));

return message;

}

@GetMapping("/users/block/{loginUserId}")

public Message blockUser(@PathVariable String loginUserId) {

Message message = new Message(this.usersService.blockUser(loginUserId));

return message;

}

@GetMapping("/users/unblock/{loginUserId}")

public Message unblockUser(@PathVariable String loginUserId) {

Message message = new Message(this.usersService.unblockUser(loginUserId));

return message;

}

@GetMapping("/users/blocked-users")

public List<Users> unblockUser() {

return this.usersService.getAllBlockedUser();

}

}

Account repository.java

package com.icinbank.repository;

import java.util.List;

import org.springframework.data.jpa.repository.JpaRepository;

import org.springframework.data.jpa.repository.Modifying;

import org.springframework.data.jpa.repository.Query;

import org.springframework.stereotype.Repository;

import com.icinbank.model.Accounts;

import com.icinbank.model.Users;

import jakarta.transaction.Transactional;

@Repository(value = "accountsRepository")

public interface AccountsRepository extends JpaRepository<Accounts, String>{

@Query(value = "SELECT \* from accounts where ACCOUNT\_NUMBER like ?1", nativeQuery = true)

List<Accounts> getAccountByAccountNumber(String accountNumber);

@Transactional

@Modifying

@Query(value = "UPDATE accounts set ACCOUNT\_IS\_NET\_BANKING\_REG=?1 where ACCOUNT\_NUMBER like ?2", nativeQuery = true)

void updateAccountNetBankingStatus(int status, String accountNumber);

@Query(value = "SELECT \* from accounts where ACCOUNT\_IS\_NET\_BANKING\_REG like ?1", nativeQuery = true)

List<Accounts> getAllUnregisteredUsers(int status);

@Transactional

@Modifying

@Query(value = "UPDATE accounts set ACCOUNT\_BALANCE\_PRIMARY=?1 where ACCOUNT\_NUMBER like ?2", nativeQuery = true)

void updatePrimaryBalance(double balance, String accountNumber);

@Transactional

@Modifying

@Query(value = "UPDATE accounts set ACCOUNT\_BALANCE\_SAVINGS=?1 where ACCOUNT\_NUMBER like ?2", nativeQuery = true)

void updateSavingsBalance(double balance, String accountNumber);

@Transactional

@Modifying

@Query(value = "UPDATE accounts set ACCOUNT\_BALANCE\_SAVINGS=?1 where ACCOUNT\_NUMBER like ?2", nativeQuery = true)

void depositSavings(int balance, String accountNumber);

@Transactional

@Modifying

@Query(value = "UPDATE accounts set ACCOUNT\_BALANCE\_PRIMARY=?1 where ACCOUNT\_NUMBER like ?2", nativeQuery = true)

void depositPrimary(int balance, String accountNumber);

@Transactional

@Modifying

@Query(value = "UPDATE accounts set ACCOUNT\_BALANCE\_SAVINGS=?1 where ACCOUNT\_NUMBER like ?2", nativeQuery = true)

void withdrawSavings(int balance, String accountNumber);

@Transactional

@Modifying

@Query(value = "UPDATE accounts set ACCOUNT\_BALANCE\_PRIMARY=?1 where ACCOUNT\_NUMBER like ?2", nativeQuery = true)

void withdrawPrimary(int balance, String accountNumber);

}

Cheque repository.java

package com.icinbank.repository;

import java.util.List;

import org.springframework.data.jpa.repository.JpaRepository;

import org.springframework.data.jpa.repository.Modifying;

import org.springframework.data.jpa.repository.Query;

import org.springframework.stereotype.Repository;

import com.icinbank.model.Cheque;

import jakarta.transaction.Transactional;

@Repository(value = "chequeRepository")

public interface ChequeRepository extends JpaRepository<Cheque, String>{

@Query(value = "SELECT \* from cheque where ACCOUNT\_NUMBER like ?1 order by CHEQUE\_BOOK\_ISSUE\_DATE desc", nativeQuery = true)

List<Cheque> getChequeBookIssuedByAccountNumber(String accountNumber);

@Query(value = "SELECT \* from cheque where CHEQUE\_BOOK\_STATUS like ?1 order by CHEQUE\_BOOK\_ISSUE\_DATE", nativeQuery = true)

List<Cheque> getAllPendingChequeBookRequests(int status);

@Transactional

@Modifying

@Query(value = "UPDATE cheque set CHEQUE\_BOOK\_STATUS=?1 where CHEQUE\_BOOK\_NUMBER like ?2", nativeQuery = true)

void updateChequeBookStatus(int status, String chequeBookNumber);

}

Transaction repository.java

package com.icinbank.repository;

import java.util.List;

import org.springframework.data.jpa.repository.JpaRepository;

import org.springframework.data.jpa.repository.Modifying;

import org.springframework.data.jpa.repository.Query;

import org.springframework.stereotype.Repository;

import com.icinbank.model.Cheque;

import jakarta.transaction.Transactional;

@Repository(value = "chequeRepository")

public interface ChequeRepository extends JpaRepository<Cheque, String>{

@Query(value = "SELECT \* from cheque where ACCOUNT\_NUMBER like ?1 order by CHEQUE\_BOOK\_ISSUE\_DATE desc", nativeQuery = true)

List<Cheque> getChequeBookIssuedByAccountNumber(String accountNumber);

@Query(value = "SELECT \* from cheque where CHEQUE\_BOOK\_STATUS like ?1 order by CHEQUE\_BOOK\_ISSUE\_DATE", nativeQuery = true)

List<Cheque> getAllPendingChequeBookRequests(int status);

@Transactional

@Modifying

@Query(value = "UPDATE cheque set CHEQUE\_BOOK\_STATUS=?1 where CHEQUE\_BOOK\_NUMBER like ?2", nativeQuery = true)

void updateChequeBookStatus(int status, String chequeBookNumber);

}