



cmpdi
A Mini Ratna Company

Central Mine Planning & Design Institute Limited

EMPLOYEE MEDICAL Bill Tracking Portal

Job Training Report

Submitted by :

Neal Chakravarty

B.Sc(CA) (23B51022)

DSPMU, RANCHI

Under the guidance of:

Mr. Saurabh Kishore, Dy. Manager

(System) CMPDI HQ, Ranchi

**DURATION OF TRAINING: 10th January 2025 to
9th February 2025**



cmpdi
A Mini Ratna Company

Central Mine Planning & Design Institute Limited

DECLARATION

I hereby declare that the job training report based on the topic **“EMPLOYEE MEDICAL Bill Tracking Portal”** submitted by me to CMPDI, Ranchi in partial fulfillment of the requirement for the award of the degree of **B.Sc(CA)** in **Dr. Shyama Prasad Mukherjee University(DSPMU)**, Ranchi is a record of bonafide industrial training undertaken by me under the supervision of **Mr. Saurabh Kishore** Dy. Manager(System), CMPDI HQ, Ranchi.

I further declare that the work reported in this report has not been submitted and will not be submitted either in part or full for the award of any other degree in this institute.

Signature of student

Neal Chakravarty

23B51022



cmpdi
A Mini Ratna Company

Central Mine Planning & Design Institute Limited

Date: 9th February 2025

Certificate

This is to certify that the job training based on **JAVA, SPRING BOOT/SPRING MVC, THYMELEAF, HTML, CSS** and **SQL** on the topic “EMPLOYEE MEDICAL Bill Tracking Portal” successfully completed by Neal Chakravarty to CMPDI, Ranchi in partial fulfillment of the requirement for the award of the degree of B.Sc(CA) is a record of bonafide job training undertaken by him under my supervision.

The training fulfills the requirement as per the regulations of this institute and in my opinion, meets the necessary standards for submission. The contents of this report have not been submitted and will not be submitted either in part or in full for the award of any other institute.

He has completed every aspect of this project with a lot of sincerity while abiding by the rules of the institution. We wish him all success for his future.

DURATION OF TRAINING: 10th January 2025 to 9th February 2025

Mr. Saurabh Kishore
Dy. Manager(System)
CMPDI HQ, Ranchi Jharkhand

Acknowledgment

In preparation for my project, I had to take the help and guidance of some respected persons, who deserve my deepest gratitude. As the completion of this project gave me much pleasure, I would like to express my special thanks of gratitude to my guide Mr. Saurabh Kishore Dy. Manager(System), CMPDI HQ for giving me a good guideline for assignment throughout numerous consultations.

I would also like to thank all those who have directly and indirectly guided me in writing this assignment.

ABOUT THE ORGANISATION

Central Mine Planning & Design Institute Limited (CMPDI) Government of India enterprise has its corporate headquarters at Ranchi in India. It is a fully owned subsidiary of Coal India Limited (CIL) and a schedule-B company a Mini Ratna (Category II) company since May 2009 and ISO 9001 certified Since March 1998, it is also on its way to ISO 27001 certification for its information security management. CMPDI functions as corporate headquarters at Ranchi and its Regional Institutes (RIS) numbered 7 are located at Asansol, Dhanbad, Ranchi, Nagpur, Bilaspur, Singrauli, and Bhubaneswar respectively along with various field units and exploration camps. So, the services of CMPDI are for any of the following purposes:

Services to pursue the company's business activities.

Services to pursue research & development needs of the industry, either independently or in association with some external agency/body. Services are technically like the above but undertaken as corporate.

CMPDI's Business Functions i.e., the consultancy and support for mineral exploration, mining, infrastructure engineering, environmental management, and management systems, especially to the mineral, mining, and allied sectors, both within and outside the coal industry and the country.

(B) CMPDI's Corporate Responsibilities, i.e., as follows

We are assisting the Ministry of Coal (MoC) and Planning Commission for strategic decisions relating to the coal sector at the national level. E.g., through maintaining inventories of coal deposits, coal mining potentials, and operations, etc.

Functioning as a nodal agency on behalf of the Government of India. E.g., for schemes funded by MoC viz S&T projects, exploration work in non-CIL blocks, Environmental Measures and Subsidence Control (EMSC) projects, and CBM clearing house, and for projects funded by CIL R&D Board.

CMPDI holds a position of eminence in the field of environment planning, impact assessment, management, and environmental engineering both in coal and other sectors. Some of the new areas in which CMPDI has diversified are environmental & hydrogeological studies for the disposal of fly ash and, the development of an air quality model.

SKILL SET BEFORE TRAINING

There are different kinds of knowledge and different ways of acquiring each kind. On one side is theory and on the other side is the practical application of theory Both types of knowledge are important. The theoretical knowledge teaches the why It helps to understand why one technique works where another fails the theoretical knowledge gained in college was very helpful in doing this training. It helps in the concepts of Building GUI and robust Applications which was the most helpful for me.

What we learned about the client's requirements and the list of procedures followed to achieve a successful product was very helpful for us, as we designed a GUI application according to the requirements of my organization. The step-wise procedure for development was useful on the other hand the types of testing at each level helped me build a successful product. Theory teaches you through the experience of others. Theoretical knowledge can often lead to a deeper understanding of a concept through seeing it in the context of a greater whole and understanding the why behind it. Whereas practical knowledge helped me acquire the specific techniques that became my strong point. It sits much closer to your actual day-to-day work. There are some things one can only learn through doing and experiencing Practical knowledge can often lead to a deeper understanding of a concept through the act of doing and personal experience. Both above are important.

It will be hard to survive in any career unless one can bring results and to do that you need practical knowledge Other concepts of programming were also very helpful while completing the project as they helped in generating logic for several activities in the project Additional learning was also very helpful for me as the knowledge gained from the course about GUI development were practically implemented Above all the most important skill of time management and being on time helped me a lot to complete the work assigned to me on time. This also helped me create a good impression on others.

KNOWLEDGE ACQUIRED FROM THE IN-PLANT TRAINING

During my one month summer internship at CMPDI Ranchi, I developed a Bill Tracking Portal for retired employees. This project enhanced my technical skills in programming (Java) and web development (Spring Boot, HTML, CSS, JavaScript). Additionally, I improved my problem-solving and communication.

The successful completion of the portal provided a reliable system for tracking bills and prepared me for future professional endeavours. This experience has been a significant stepping stone in my career journey.

LANGUAGES/TECH USED-

EMPLOYEE MEDICAL Bill Tracking Portal

Language Used	HTML,CSS, BOOTSTRAP, SQL, JAVA
Framework Used	SPRING BOOT
Project Type	Web Application

HTML:

- HTML stands for Hyper Text Markup Language.
- HTML is the standard markup language for creating Web pages.
- HTML describes the structure of a Web page.
- HTML consists of a series of elements.
- HTML elements tell the browser how to display the content.
- HTML elements label pieces of content such as “this is a heading”, “this is a paragraph”, “this is a link”, etc.

CSS:

- CSS stands for Cascading Style Sheets.
- CSS describes how HTML elements are to be displayed on screen, paper, or in other media.
- CSS saves a lot of work. It can control the layout of multiple web pages all at once.
- External stylesheets are stored in CSS files.

BOOTSTRAP:

- BOOTSTRAP has responsive grid system for layout.
- BOOTSTRAP has pre-styled UI components (buttons, forms, navigation bars).
- BOOTSTRAP has built-in CSS classes for quick styling.
- BOOTSTRAP has JavaScript plugins for interactive components (modals, carousels).
- BOOTSTRAP has customizable themes and variables.
- BOOTSTRAP has extensive documentation and community support

SQL:

- SQL stands for Structured Query Language.
- SQL is used for managing and manipulating relational databases.
- SQL provides a standardized way to interact with databases, perform data operations, and retrieve information.
- SQL enables to combine data from multiple tables using data joins.
- SQL allows to creation of virtual tables called view

ABSTRACT

The project report outlines the development and implementation of the "CMPDI CPRMSE/NE Bill Tracking Portal," a comprehensive solution tailored for Central Mine Planning and Design Institute Limited (CMPDI) .This EMPLOYEE MEDICAL Bill Tracking Portal is designed with a dual-interface system to cater to different user roles: the User Side and the Admin Side.

User Side: The User Side interface is designed for retired employees, providing them with an intuitive and accessible platform to track their bill statuses/Transactions. Key functionalities include:

- User Authentication: Secure login and registration.
- Bill View: Overview of all the transactions.

Admin Side: The Admin Side interface is tailored for administrators to manage and oversee the bill tracking process efficiently. Key functionalities include:

- Admin Authentication: Secure login for administrators.
- Report View: Tools for mapping and viewing transactions by Employee-code, Month, year independently and also by combining any two attributes out of three.
- Add Transaction : An utility to upload bulk transaction enteries.
- Update Details: A complete utility for admins to update particular fields if required.

By structuring the system in this way, the portal ensures that both retired employees and administrators have access to the functionalities they need, enhancing overall efficiency and user satisfaction.

OBJECTIVE OF THE PROJECT

The primary objective of the "EMPLOYEE MEDICAL Bill Tracking Portal" project is to develop and implement a robust, user-friendly system that streamlines the bill tracking process for retired employees and administrators at Central Mine Planning and Design Institute Limited (CMPDI). The portal aims to:

1. **Enhance Accessibility and User Experience:** Provide retired employees with an intuitive platform to securely track their bill statuses and transactions, ensuring ease of use and accessibility.
2. **Improve Administrative Efficiency:** Equip administrators with comprehensive tools to manage and oversee the bill tracking process efficiently, including secure authentication, detailed report views, bulk transaction uploads, and update utilities.

By achieving these objectives, the project seeks to improve overall efficiency, user satisfaction, and data management within CMPDI, benefiting both retired employees and administrative staff.

INTERFACES

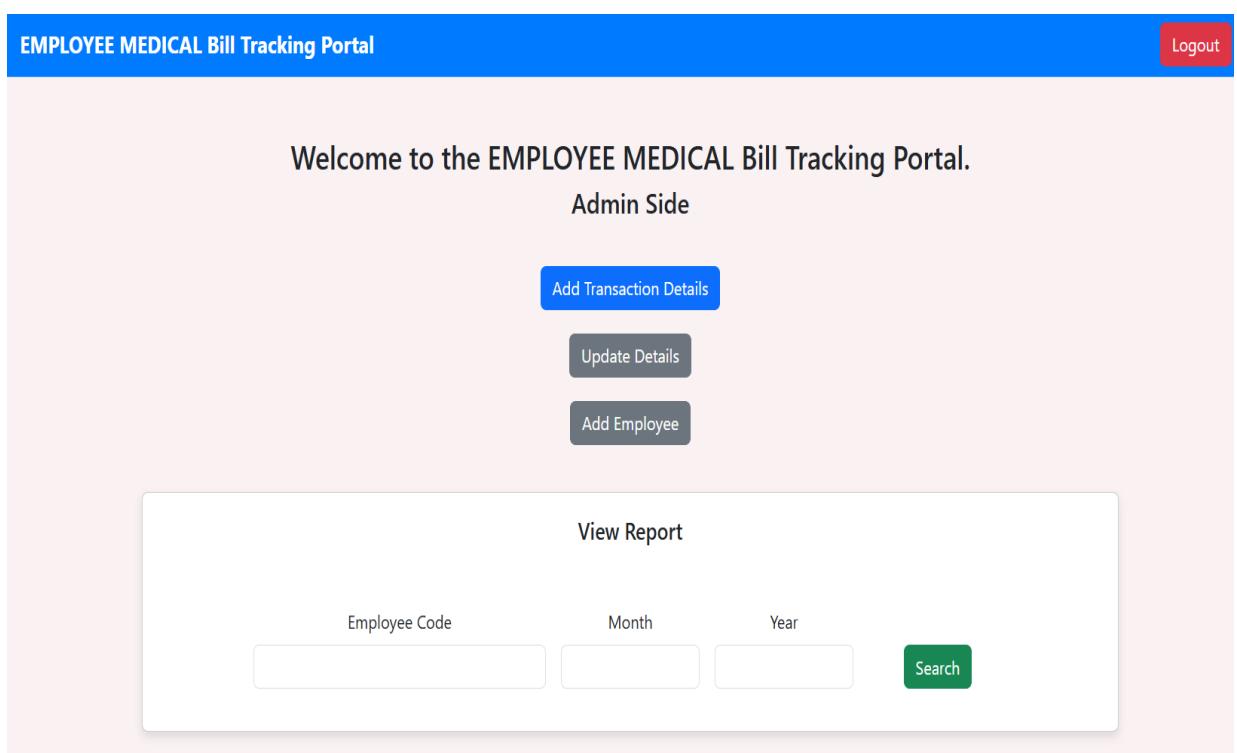
The entire project mainly consists of 3 interfaces:

1) Home page:



(a common login page)->

2) Admin page:



3) user page:

EMPLOYEE MEDICAL Bill Tracking Portal

Logout

Employee/Claimant Details

Employee Code: EMP001
Employee Name: Shaksham
DOB: 02-10-2000
Aadhar: 123456789123
Location: R5
Medical Card No.: 4574698438/4
Claiment Name: Samay
Claiment Aadhar: 7484859894646894

Medical Claim Details

Search Search

Year	Month	Claimed	Passed	Remarks
2005	2	792	496	OPD
1999	12	49626	7898	OPD
1999	5	7896	1258785	OPD
1999	4	79632	14595	OPD
2015	10	585	789	OPD

EMPLOYEE MEDICAL Bill Tracking Portal

Logout

Employee/Claimant Details

Employee Code: EMP001
Employee Name: Shaksham
DOB: 02-10-2000
Aadhar: 123456789123
Location: R5
Medical Card No.: 4574698438/4
Claiment Name: Samay
Claiment Aadhar: 7484859894646894

Medical Claim Details

Search Search

Year	Month	Claimed	Passed	Remarks
2005	2	792	496	OPD
1999	12	49626	7898	OPD
1999	5	7896	1258785	OPD
1999	4	79632	14595	OPD
2015	10	585	789	OPD

BREIFING

- ADMIN PAGE
- USER PAGE

Admin Page:

The admin page can be accessed by employees with role "ADMIN" after successful authentication, the authentication is being managed by spring security and by the data present in the associated database, the admin has been given a variety of utilities or modules to work with USER data and ensure effective management.

There are four utilities for the ADMIN, these are as follows ->

- **VIEW REPORT:** A report viewing probation for the ADMN to view transactions by filtering transaction by 'employee-code', 'month', 'year' all three attributes can be used independently as well as in combination to fetch desired transaction details.

View Report

Employee Code Month Year

Search

Transaction Details

Month	Year	Claimed	Passed	Remarks
2	2005	792	496	OPD
12	1999	49626	7898	OPD
5	1999	7896	1258785	OPD
4	1999	79632	14595	OPD
10	2015	585	789	OPD

- **ADD TRANSACTION DETAILS** : An utility page for ADMIN's to upload transactional entries to the database in bulk by just uploading the ',CSV' file to the upload element in the page . This is a robust functionality to overcome the manual data entry work or query writing making the management of transactions easier for the ADMINS' and tis saves a lot of time. This feature uploads all the transactions/entries to the database efficiently irrespective of the number of entries.

Home Logout

Upload CSV File

Attach the file below

Drag file(s) here to upload.
Alternatively, you can select a file by clicking here.

Upload

- **UPDATE DETAILS :** An utility page for ADMIN's to update columns like 'Claiment Name' and 'Claiment Aadhar' , as it is a bill tracking portal and a lot of transactions are required to be managed data hinderence is possible therefore an update details utility stands out to manage such hinderences and ensure data consistency.

The screenshot shows the 'EMPLOYEE MEDICAL Bill Tracking Portal' Admin Side. At the top, there are 'Logout' and 'Home' buttons. Below that, a welcome message reads 'Welcome to the EMPLOYEE MEDICAL Bill Tracking Portal. Admin Side'. A search bar contains the placeholder 'Search by Employee Code...'. A blue 'Search' button is below it. The main area is titled 'Employee Details' and displays the following information for employee Aman:

Employee Code:	EMP002
Employee Name:	Aman
Employee Aadhar:	123456789132
Date of Birth:	02-02-1987
Claiment Name:	Aman
Claiment Aadhar:	29876812746
Medical Card No:	12486549597
Role:	USER

- **ADD EMPLOYEE :** A simple registration page for ADMINS to add employee to the database.

The screenshot shows the 'Add Employee Form' interface. It consists of several input fields with labels and icons:

- Employee Code: Input field with placeholder 'Enter Employee Code'.
- Name: Input field with placeholder 'Enter Name'.
- Date of Birth: Input field with placeholder 'dd-mm-yyyy' and a calendar icon.
- Aadhar: Input field with placeholder 'Enter Aadhar Number'.
- Claiment Name: Input field with placeholder 'Enter Claiment Name'.
- Claiment Aadhar: Input field with placeholder 'Enter Claiment Aadhar Number'.
- Location: Input field with placeholder 'Enter Location'.
- Medical Card No: Input field with placeholder 'Enter Medical Card No' containing 'EMP003'.
- Password: Input field with placeholder 'Enter Password' containing '*****'.

At the bottom right is a large blue 'Register' button.

LOGIN FORM: Role And Authorization

The Form gets displayed as "/login" request is hit , the login form is common for both USER and ADMIN.

The form is a mobile-style login screen. At the top center, it says "Log In". Below that is a text input field labeled "Username" with the placeholder "Employee Code". Below that is a text input field labeled "Password" with the placeholder "Aadhar". At the bottom is a large blue button labeled "Sign in".

Key Features

- ⊕ The Form accepts Employee Code as the username and Aadhar associated to that Employee Code as the password.
- ⊕ Use of Role-based authentication allows both USER and ADMIN to access the same login page and as per their role get redirected to their respective pages.
- ⊕ BY the help of backend logic the Employee Code entered in the username field is saved in session to fetch details(for USER) and maintain some specific records if updates carried out.

INSIGHTS TO THE SOURCE CODE

Here are some screenshots of the project folder/file structure
And codes of what the project looks like.-----

The screenshot shows the file structure of the 'CPRMSE-NE-poratl' project. The root directory contains several sub-folders and files:

- src/main/resources**: Contains static files and templates.
 - static
 - templates
 - ademp.html
 - admin.html
 - crudutil.html
 - home.html
 - login.html
 - logout.html
 - upload.html
 - values.html
 - application.properties
- JRE System Library [JavaSE-17]
- Maven Dependencies
- src/main/java**: Contains Java source code for various packages.
 - com.CMPDI.CPRMSE.NE.poratl
 - Application.java
 - com.CMPDI.CPRMSE.NE.poratl.config
 - SecurityConfig.java
 - com.CMPDI.CPRMSE.NE.poratl.controller
 - AdminController.java
 - HomeController.java
 - com.CMPDI.CPRMSE.NE.poratl.models
 - AppUser.java
 - Transactions.java
 - com.CMPDI.CPRMSE.NE.poratl.repositories
 - AppUserRepository.java
 - TransactionsRepository.java
 - com.CMPDI.CPRMSE.NE.poratl.Response
 - ResponseMessage.java
 - com.CMPDI.CPRMSE.NE.poratl.services
 - AppUserService.java
 - Fileservice.java
 - Fileserviceimpl.java
 - TransactionService.java
 - com.CMPDI.CPRMSE.NE.poratl.util
 - CustomAuthenticationSuccessHandler.java
 - PasswordEncodingScript.java
- src/test/java**
- SRC**
- target**: Contains build artifacts.
 - HELP.md
 - mvnw
 - mvnw.cmd
 - pom.xml

Structural visualization:

```
CPRMSE-NE-portal
├── src
│   └── main
│       ├── java
│       │   └── com.CMPDI.CPRMSE.NE.portal
│       │       ├── Application.java
│       │       ├── SecurityConfig.java
│       │       └── controller
│       │           ├── AdminController.java
│       │           └── HomeController.java
│       ├── models
│       │   ├── AppUser.java
│       │   └── Transactions.java
│       ├── repositories
│       │   ├── AppUserRepository.java
│       │   └── TransactionsRepository.java
│       ├── Response
│       │   └── ResponseMessage.java
│       ├── services
│       │   ├── AppUserService.java
│       │   ├── Fileservice.java
│       │   ├── Fileserviceimpl.java
│       │   └── TransactionService.java
│       └── util
│           ├── CustomAuthenticationSuccessHandler.java
│           └── PasswordEncodingScript.java
└── resources
    ├── static
    └── templates
        ├── admin.html
        ├── crudutil.html
        ├── home.html
        ├── logout.html
        ├── upload.html
        └── values.html
└── target (not expanded)
└── src (another source directory not expanded)
└── HELP.md
└── mvnw
└── mvnw.cmd
└── pom.xml
```

- *HTML/CSS(thymeleaf) templates:*

eclipse-workspace - CPRMSE-NE-portat/src/main/resources/templates/home.html - Eclipse IDE

File Edit Source Navigate Search Project Run Window Help

Project Explorer X

Fileservice... Transaction... Fileservice... Transaction... Application... AdminControl... SecurityConf... crudit.html admin.html home.html

bank

CPRMSE-NE-portat (in CPRMSE-NE-portat (2))

src/main/resources

static

templates

- addemp.html
- admin.html
- crudit.html
- home.html
- login.html
- logout.html
- upload.html
- values.html

application.properties

JRE System Library [JavaSE-17]

Maven Dependencies

src/main/java

- com.CMPDLCPRMSE.NE.portat
- Application.java
- com.CMPDLCPRMSE.NE.portat.config
- SecurityConfig.java
- com.CMPDLCPRMSE.NE.portat.controller
- AdminController.java
- HomeController.java
- com.CMPDLCPRMSE.NE.portat.models
- AppUser.java
- Transactions.java
- com.CMPDLCPRMSE.NE.portat.repositories
- AppUserRepository.java
- TransactionsRepository.java
- com.CMPDLCPRMSE.NE.portat.Response
- ResponseMessage.java
- com.CMPDLCPRMSE.NE.portat.services
- AppUserService.java
- Fileservice.java
- Fileserviceimpl.java
- TransactionService.java
- com.CMPDLCPRMSE.NE.portat.util

Fileservice... Transaction... Fileservice... Transaction... Application... AdminControl... SecurityConf... crudit.html admin.html home.html

```
<!DOCTYPE html>
<html lang="en" xmlns:th="http://www.thymeleaf.org">
    <head>
        <meta charset="utf-8">
        <meta name="viewport" content="width=device-width, initial-scale=1">
        <title>CPRMSE/NE</title>
        <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-Q&lt;...&gt;">
        <style>
            body {
                font-family: Arial, sans-serif;
                background-size: cover;
                color: #333;
                background-color: #f9f1f2;
            }
            .navbar {
                background-color: #007bff;
            }
            .navbar-brand {
                font-weight: bold;
                color: #fff !important;
            }
            .nav-link{
                color: #fff !important;
            }
            .container {
                max-width: 1200px;
            }
            .main-content {
                background-color: rgba(255, 255, 255, 0.9);
                border-radius: 10px;
                padding: 30px;
                margin-top: 20px;
                box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
            }
        </style>
    </head>
    <body>
```

Problems Javadoc Console Debug

No consoles to display at this time.

Writable Insert 1:1:0

22°C Haze ENG IN 09-03-2025

eclipse-workspace - CPRMSE-NE-portal/src/main/resources/templates/home.html - Eclipse IDE

File Edit Source Navigate Search Project Run Window Help

Project Explorer X

Fileservice... Transaction... Fileservice... Transaction... Application... AdminControl... SecurityConf... crudutil.html admin.html home.html X

bank

✓ CPRMSE-NE-portal (in CPRMSE-NE-portal (2))

src/main/resources

- static
- templates
 - ademp.html
 - admin.html
 - crudutil.html
 - home.html
 - login.html
 - logout.html
 - upload.html
 - values.html
- application.properties

JRE System Library [JavaSE-17]

Maven Dependencies

src/main/java

- com.CMPDLCPRMSE-portal
- Application.java
- com.CMPDLCPRMSE-portal.config
- SecurityConfig.java
- com.CMPDLCPRMSE-portal.controller
- AdminController.java
- HomeController.java
- com.CMPDLCPRMSE-portal.models
- AppUser.java
- Transactions.java
- com.CMPDLCPRMSE-portal.repositories
- AppUserRepository.java
- TransactionsRepository.java
- com.CMPDLCPRMSE-portal.Response
- ResponseMessage.java
- com.CMPDLCPRMSE-portal.services
- AppUserService.java
- Fileservice.java
- Fileserviceimpl.java
- TransactionService.java

com.CMPDLCPRMSE-portal.util

79 <body>

80 <nav class="navbar navbar-expand-lg border-bottom">

81 <div class="container">

82 CPRMSE/NE Bill Tracking Portal

83 <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent" aria-

84

85 </button>

86 <div class="collapse navbar-collapse" id="navbarSupportedContent">

87 <ul class="navbar-nav me-auto mb-2 mb-lg-0">

88 <li class="nav-item">

89 Login

90

91

92 </div>

93 </div>

94 </div>

95 </nav>

96 <div class="container">

97

98 <div id="content">

99 <div class="photobanner">

100

101

102

103

104

105 <!-- Duplicate images for continuous scrolling -->

106

107

108

109

110

111 </div>

Problems Javadoc Console Debug

No consoles to display at this time.

Writable Insert 1:1:0

22°C Haze

Search

ENG IN

2039

09-03-2025

• SECURITY CONFIGURATIONS:

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project structure under "CPRMSE-NE-poratl".
- Code Editor:** Displays the `SecurityConfig.java` file content.
- Toolbar:** Includes standard Eclipse icons for File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, Help.
- Bottom Status Bar:** Shows "Writable", "Smart Insert", "42:5:1656", and system icons.

```
12
13 @Configuration
14 @EnableWebSecurity
15 public class SecurityConfig {
16
17     @Bean
18     public SecurityFilterChain securityFilterChain(HttpSecurity http) throws Exception {
19         return http.authorizeHttpRequests(auth -> auth
20             .requestMatchers("/").permitAll()
21             .requestMatchers("/logout").permitAll()
22             .requestMatchers("/images/**").permitAll()
23             .requestMatchers("/admin/**").hasRole("ADMIN")
24             .requestMatchers("/user/**").hasRole("USER")
25             .anyRequest().authenticated()
26         )
27             .formLogin(form -> form
28                 .loginPage("/Login").permitAll()
29             )
30             .successHandler(new CustomAuthenticationSuccessHandler())
31         )
32         .logout(config -> config.logoutSuccessUrl("/"))
33         .build();
34     }
35
36     @Bean
37     public PasswordEncoder passwordEncoder() {
38         return new BCryptPasswordEncoder();
39     }
40
41
42
43 }
44
```

Bottom status bar: 22°C, Haze, ENG IN, 09-03-2025, 20:49

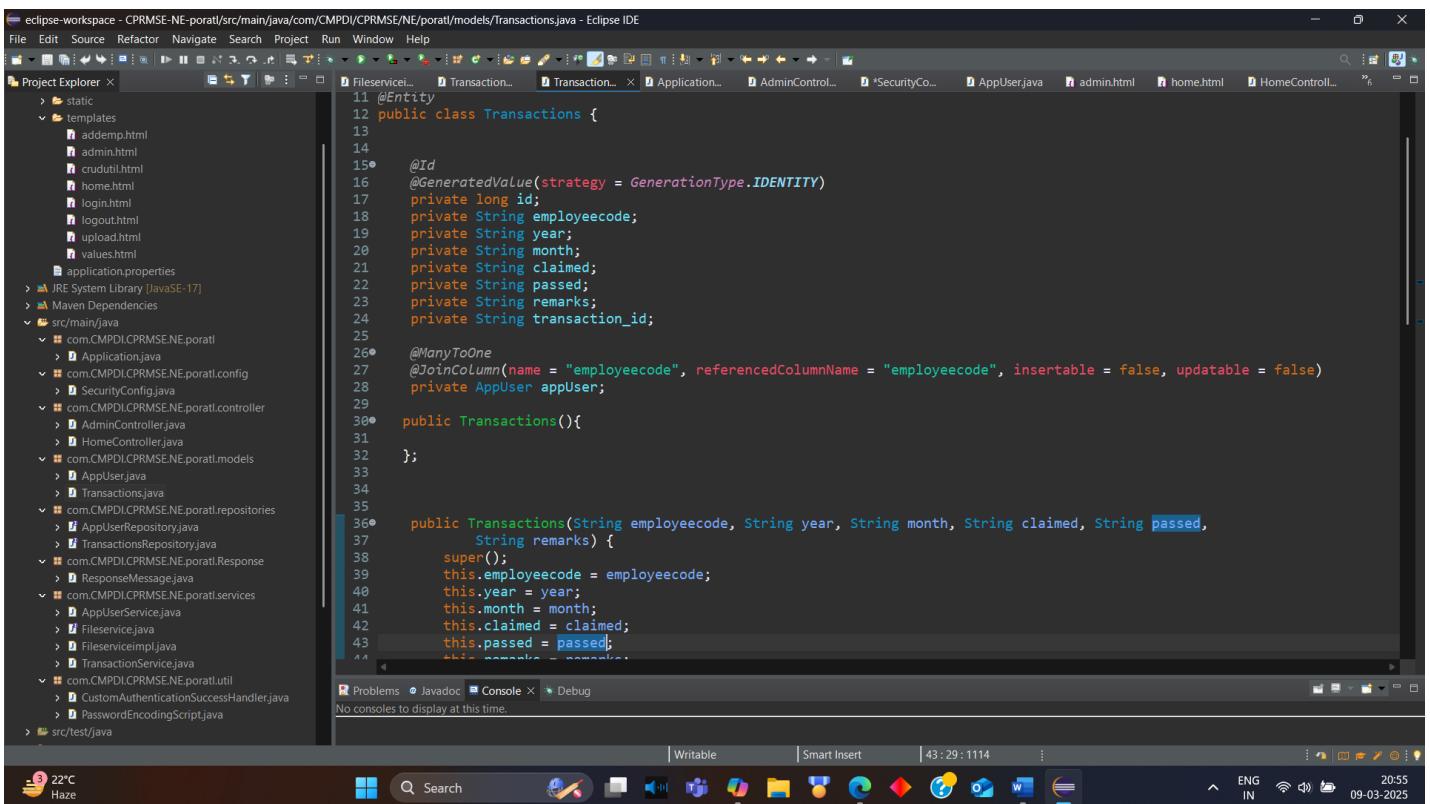
- **CONTROLLERS FOLDER:**

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer View:** Shows the project structure. The `src/main/java` folder contains several packages:
 - `com.CMPDI.CPRMSE.NE.poratl`: Contains `Application.java`, `SecurityConfig.java`, `AdminController.java`, `HomeController.java`, `AppUser.java`, `Transactions.java`, `AppUserRepository.java`, `TransactionsRepository.java`, `ResponseMessage.java`, `AppUserService.java`, `Fileservice.java`, `Fileserviceimpl.java`, and `Transactionservice.java`.
 - `com.CMPDI.CPRMSE.NE.poratl.Response`: Contains `ResponseMessage.java`.
 - `com.CMPDI.CPRMSE.NE.poratl.services`: Contains `AppUserService.java`, `Fileservice.java`, `Fileserviceimpl.java`, and `Transactionservice.java`.
- Code Editor View:** Displays the `AdminController.java` file with the following code:

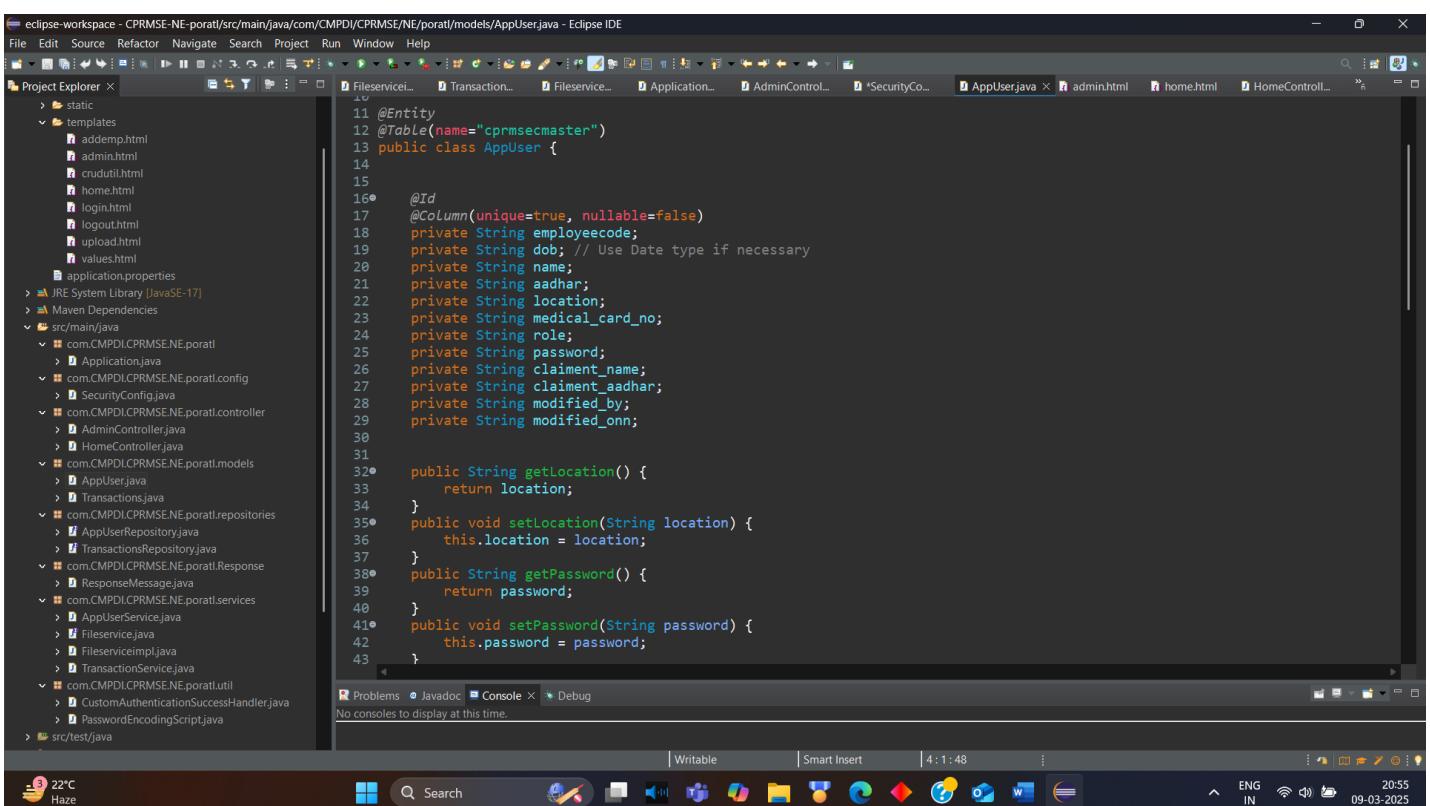
```
31 package com.CMPDI.CPRMSE.NE.poratl.controller;
32 
33 public class AdminController {
34     @Autowired
35     private AppUserService appUserService;
36 
37     @Autowired
38     private Fileservice service;
39 
40     @Autowired
41     private TransactionService transactionService;
42 
43     @Autowired
44     private PasswordEncoder passwordEncoder;
45 
46     @GetMapping("/admin")
47     public String adminPage(HttpSession session, Model model) {
48         String employeeCode = (String) session.getAttribute("employeeCode");
49 
50         if (employeeCode != null) {
51             AppUser appUser = appUserService.findByEmployeecode(employeeCode);
52             model.addAttribute("appUser", appUser);
53         }
54 
55         return "admin";
56     }
57 
58     @GetMapping("/admin/crudutil")
59     public String crudutil() {
60         return "crudutil";
61     }
62 }
```
- Bottom Status Bar:** Shows the system tray with icons for battery, signal strength, and network.

• ENTITY/MODELS FOLDER:



The screenshot shows the Eclipse IDE interface with the project 'eclipse-workspace - CRMSE-NE-portal' open. The 'Transactions.java' file is displayed in the central editor area. The code defines a Java class 'Transactions' with annotations for Entity and Id. It includes fields for employee code, year, month, claimed status, passed status, and a transaction ID, along with a many-to-one relationship to 'AppUser'. The code is well-formatted with color-coded syntax highlighting.

```
11 @Entity
12 public class Transactions {
13
14
15•     @Id
16     @GeneratedValue(strategy = GenerationType.IDENTITY)
17     private long id;
18     private String employeecode;
19     private String year;
20     private String month;
21     private String claimed;
22     private String passed;
23     private String remarks;
24     private String transaction_id;
25
26•     @ManyToOne
27     @JoinColumn(name = "employeecode", referencedColumnName = "employeecode", insertable = false, updatable = false)
28     private AppUser appUser;
29
30•     public Transactions(){}
31
32• }
33
34
35
36•     public Transactions(String employeecode, String year, String month, String claimed, String passed,
37     String remarks) {
38     super();
39     this.employeecode = employeecode;
40     this.year = year;
41     this.month = month;
42     this.claimed = claimed;
43     this.passed = passed;
44 }
```



The screenshot shows the Eclipse IDE interface with the project 'eclipse-workspace - CRMSE-NE-portal' open. The 'AppUser.java' file is displayed in the central editor area. The code defines a Java class 'AppUser' with annotations for Entity and Table. It includes fields for employee code, date of birth, name, Aadhar, location, medical card number, role, password, claiment name, and modified details. The code is well-formatted with color-coded syntax highlighting.

```
11 @Entity
12 @Table(name="cprmsecmaster")
13 public class AppUser {
14
15
16•     @Id
17     @Column(unique=true, nullable=false)
18     private String employeecode;
19     private String dob; // Use Date type if necessary
20     private String name;
21     private String aadhar;
22     private String location;
23     private String medical_card_no;
24     private String role;
25     private String password;
26     private String claiment_name;
27     private String claiment_aadhar;
28     private String modified_by;
29     private String modified_onn;
30
31
32•     public String getLocation() {
33         return location;
34     }
35•     public void setLocation(String location) {
36         this.location = location;
37     }
38•     public String getPassword() {
39         return password;
40     }
41•     public void setPassword(String password) {
42         this.password = password;
43     }
44 }
```

• REPOSITORY LAYER:

eclipse-workspace - CPRMSE-NE-portal/src/main/java/com/CMPDI/CPRMSE/NE/portal/repositories/TransactionsRepository.java - Eclipse IDE

```

File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer X Transaction... X Transaction... AdminControl... *SecurityCo... AppUser.java admin.html home.html HomeControll... AppUserRepo... >
src/main/java
  com.CMPDI.CPRMSE.NE.portal
    Application.java
  com.CMPDI.CPRMSE.NE.portal.config
    SecurityConfig.java
  com.CMPDI.CPRMSE.NE.portal.controller
    AdminController.java
    HomeController.java
  com.CMPDI.CPRMSE.NE.portal.models
    AppUser.java
    Transactions.java
  com.CMPDI.CPRMSE.NE.portal.repositories
    AppUserRepository.java
    TransactionsRepository.java
  com.CMPDI.CPRMSE.NE.portal.Response
    ResponseMessage.java
  com.CMPDI.CPRMSE.NE.portal.services
    AppUserService.java
    Fileservice.java
    Fileserviceimpl.java
    TransactionService.java
  com.CMPDI.CPRMSE.NE.portal.util
    CustomAuthenticationSuccessHandler.java
    PasswordEncodingScript.java
src/test/java
src
target
HELP.md
mvnw
mvnw.cmd
pom.xml

```

```

58
59
60 import java.util.List;□
61
62 @Repository
63 public interface TransactionsRepository extends JpaRepository<Transactions, Long> {
64
65     public List<Transactions> findTransactionsByEmployeecode(String employeecode);
66
67     @Query("SELECT t FROM Transactions t WHERE t.employeecode = :employeeCode"
68           + " AND (:month IS NULL OR t.month = :month) " + " AND (:year IS NULL OR t.year = :year)")
69     List<Transactions> findByEmployeecodeAndOptionalMonthAndYear(@Param("employeeCode") String employeeCode,
70                                                               @Param("month") String month, @Param("year") String year);
71
72     List<Transactions> findByEmployeecodeAndMonth(@Param("employeecode") String employeecode,
73                                                   @Param("month") String month);
74
75     List<Transactions> findByEmployeecodeAndYear(@Param("employeecode") String employeecode,
76                                                 @Param("year") String year);
77
78     List<Transactions> findByMonthAndYear(@Param("month") String month,
79                                           @Param("year") String year);
80
81     List<Transactions> findByEmployeecode(@Param("employeecode") String employeecode);
82
83     List<Transactions> findByMonth(@Param("month") String month);
84
85     List<Transactions> findByYear(@Param("year") String year);
86
87 }
88
89
90
91
92
93
94
95 }
96
97

```

Problems Javadoc Console Debug

No consoles to display at this time.

Writable Smart Insert | 94:5:3429

IND - NZ Live Search File Explorer View Window Help ENG IN 20:58 09-03-2025

eclipse-workspace - CPRMSE-NE-portal/src/main/java/com/CMPDI.CPRMSE.NE.portal.repositories/AppUserRepository.java - Eclipse IDE

```

File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer X Transaction... X Transaction... AdminControl... *SecurityCo... AppUser.java admin.html home.html HomeControll... AppUserRepo... >
src/main/java
  com.CMPDI.CPRMSE.NE.portal
    Application.java
  com.CMPDI.CPRMSE.NE.portal.config
    SecurityConfig.java
  com.CMPDI.CPRMSE.NE.portal.controller
    AdminController.java
    HomeController.java
  com.CMPDI.CPRMSE.NE.portal.models
    AppUser.java
    Transactions.java
  com.CMPDI.CPRMSE.NE.portal.repositories
    AppUserRepository.java
    TransactionsRepository.java
  com.CMPDI.CPRMSE.NE.portal.Response
    ResponseMessage.java
  com.CMPDI.CPRMSE.NE.portal.services
    AppUserService.java
    Fileservice.java
    Fileserviceimpl.java
    TransactionService.java
  com.CMPDI.CPRMSE.NE.portal.util
    CustomAuthenticationSuccessHandler.java
    PasswordEncodingScript.java
src/test/java
src
target
HELP.md
mvnw
mvnw.cmd
pom.xml

```

```

1 package com.CMPDI.CPRMSE.NE.portal.repositories;
2
3 import java.util.List;□
4
5 @Repository
6 public interface AppUserRepository extends JpaRepository<AppUser, String> {
7
8     public AppUser findByEmployeecode(String employeecode);
9
10    public @NotNull List<AppUser> findAll();
11
12 }
13
14
15

```

Problems Javadoc Console Debug

No consoles to display at this time.

Writable Smart Insert | 1:1:0

Sports headline Mick Schumacher... Search File Explorer View Window Help ENG IN 20:59 09-03-2025

SERVICE LAYER:

eclipse-workspace - CPRMSE-NE-portal/src/main/java/com/CMPDI/CPRMSE/NE/portal/services/Fileserviceimpl.java - Eclipse IDE

```

File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer X Fileservice... X Transaction... X Fileservice... X Transaction... X AdminControl... X AppUser.java X HomeControll... X AppUserRepo... X TransactionS...
118
119 @Service
120 public class Fileserviceimpl implements Fileservice {
121
122     @Autowired
123     private TransactionsRepository repo;
124
125     @Override
126     public boolean hasCsvFormat(MultipartFile file) {
127         String type = "text/csv";
128         if (!type.equals(file.getContentType())) {
129             return false;
130         }
131         return true;
132     }
133
134     @Override
135     public void ProcessAndSaveData(MultipartFile file) {
136         try {
137             List<Transactions> transactions = csvToTransactions(file.getInputStream());
138             repo.saveAll(transactions);
139         } catch (IOException e) {
140             // TODO Auto-generated catch block
141             e.printStackTrace();
142         }
143     }
144
145     private List<Transactions> csvToTransactions(InputStream inputStream) {
146         try (BufferedReader fileReader = new BufferedReader(new InputStreamReader(inputStream, "UTF-8"));
147              CSVParser csvParser = new CSVParser(fileReader,
148                      CSVFormat.DEFAULT.withFirstRecordAsHeader().withIgnoreHeaderCase().withTrim()) {
149             // processing the CSV data
150             List<Transactions> transactions = new ArrayList<>();

```

Problems Javadoc Console Debug
No consoles to display at this time.

Writable Smart Insert 172:1 7:104

22°C Haze ENG IN 21:05 09-03-2025

eclipse-workspace - CPRMSE-NE-portal/src/main/java/com/CMPDI/CPRMSE/NE/portal/services/TransactionService.java - Eclipse IDE

```

File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer X Fileservice... X Transaction... X Fileservice... X Transaction... X AdminControl... X AppUser.java X HomeControll... X AppUserRepo... X TransactionS...
1 package com.CMPDI.CPRMSE.NE.portal.services;
30 * import org.springframework.beans.factory.annotation.Autowired; import[]
20
21 import org.springframework.beans.factory.annotation.Autowired;[]
28
29 @Service
30 public class TransactionService {
31
32     @Autowired
33     private TransactionsRepository transactionsRepository;
34
35     public List<Transactions> getTransactions(String employeecode, String month, String year) {
36         if (month != null && !month.isEmpty() && year != null && !year.isEmpty()) {
37             return transactionsRepository.findByEmployeecodeAndOptionalMonthAndYear(employeecode, month, year);
38         } else if (month != null && !month.isEmpty() && employeecode != null && !employeecode.isEmpty()) {
39             return transactionsRepository.findByEmployeecodeAndMonth(employeecode, month);
40         } else if (year != null && !year.isEmpty() && employeecode != null && !employeecode.isEmpty()) {
41             return transactionsRepository.findByEmployeecodeAndYear(employeecode, year);
42         }
43         else if ((employeecode == null || employeecode.isEmpty()) && year != null && !year.isEmpty() && month != null && !month.isEmpty())
44             return transactionsRepository.findByMonthAndYear(month, year);
45
46         else if ((employeecode == null || employeecode.isEmpty()) && (year == null || year.isEmpty()) && month != null && !month.isEmpty())
47             return transactionsRepository.findByMonth(month);
48
49         else if ((employeecode == null || employeecode.isEmpty()) && (month == null || month.isEmpty()) && year != null && !year.isEmpty())
50             return transactionsRepository.findByYear(year);
51
52         else {
53             return transactionsRepository.findByEmployeecode(employeecode);
54         }
55     }
56

```

Problems Javadoc Console Debug
No consoles to display at this time.

Writable Smart Insert 1:1:0

22°C Haze ENG IN 21:06 09-03-2025

• UTILITY LAYER:

The screenshot shows the Eclipse IDE interface with the title bar "eclipse-workspace - CPRMSE-NE-porati/src/main/java/com/CMPDI/CPRMSE/NE/porati/util/CustomAuthenticationSuccessHandler.java - Eclipse IDE". The left side features the Project Explorer view displaying the project structure. The right side shows the code editor with the following Java code:

```
1 package com.CMPDI.CPRMSE.NE.porati.util;
2
3 * import org.springframework.security.core.Authentication; import
4
5 public class CustomAuthenticationSuccessHandler implements AuthenticationSuccessHandler {
6
7     @Override
8     public void onAuthenticationSuccess(HttpServletRequest request, HttpServletResponse response, Authentication authentication) {
9         // Assuming employeeCode is the username
10        String employeeCode = authentication.getName();
11
12        // Store employee code in session
13        HttpSession session = request.getSession();
14        session.setAttribute("employeeCode", employeeCode);
15
16        // Check the role and redirect accordingly
17        Collection< GrantedAuthority> authorities = authentication.getAuthorities();
18        for (GrantedAuthority authority : authorities) {
19            if (authority.getAuthority().equals("ROLE_ADMIN")) {
20                response.sendRedirect("/admin");
21                return;
22            } else if (authority.getAuthority().equals("ROLE_USER")) {
23                response.sendRedirect("/values");
24                return;
25            }
26        }
27
28        // Default redirect
29        response.sendRedirect("/");
30    }
31
32 }
```

The status bar at the bottom shows system information like "22°C Haze" and "09-03-2025".

The screenshot shows the Eclipse IDE interface with the title bar "eclipse-workspace - CPRMSE-NE-porati/src/main/java/com/CMPDI/CPRMSE/NE/porati/util/PasswordEncodingScript.java - Eclipse IDE". The left side features the Project Explorer view displaying the project structure. The right side shows the code editor with the following Java code:

```
1 package com.CMPDI.CPRMSE.NE.porati.util;
2
3 import org.springframework.beans.factory.annotation.Autowired;
4
5 @Component
6 public class PasswordEncodingScript implements CommandLineRunner {
7
8     private static final Logger logger = LoggerFactory.getLogger(PasswordEncodingScript.class);
9
10    @Autowired
11    private AppUserRepository appUserRepository;
12
13    @Autowired
14    private PasswordEncoder passwordEncoder;
15
16    @Override
17    public void run(String... args) throws Exception {
18        Iterable<AppUser> users = appUserRepository.findAll();
19
20        for (AppUser user : users) {
21            if (!user.getPassword().startsWith("$2a$")) { // Ensure encoding is done only for non-encoded passwords
22                String encodedPassword = passwordEncoder.encode(user.getAadhar());
23                user.setPassword(encodedPassword);
24                appUserRepository.save(user);
25                logger.info("Encoded password for user: " + user.getEmployeecode());
26            }
27        }
28    }
29
30 }
```

The status bar at the bottom shows system information like "22°C Haze" and "09-03-2025".

Scope of the Project:

The EMPLOYEE MEDICAL Bill Tracking Portal is an innovative web-based solution designed to streamline the process of bill tracking and management. This portal aims to serve the needs of organizations and their stakeholders by offering two intuitive interfaces: one for administrators and another for general users. By leveraging cutting-edge technologies and a user-centric design, this portal enhances the overall efficiency and transparency of bill processing.

Administrator Interface:

The administrator interface is tailored to meet the needs of system administrators who are responsible for managing bill-related operations. This interface provides comprehensive tools for managing various aspects of bill tracking:

1. View Reports: Administrators can generate reports based on employee code, month, and year. This feature allows for detailed tracking and analysis of transactions.
2. Add Transaction Detail: Administrators can upload bulk transaction entries to the database, streamlining data entry and ensuring accuracy.
3. Update Details: Administrators can update specific fields in the database as required, maintaining up-to-date and accurate information.
4. Add Employee: Administrators have the ability to add new employees to the master table of the database, ensuring the system is up-to-date with the latest personnel information.

User Interface:

The user interface is designed to offer a seamless and user-friendly experience for general users. Users can easily submit bills, track their status in real-time, and receive notifications for any updates. The interface provides a clear and organized view of all submitted bills, allowing users to search, filter, and sort bills based on various criteria. By offering transparency and real-time updates, the portal empowers users to stay informed and engaged with their bill management process. Additionally, users can view their reports of transactions, providing them with an overview of their bill history.

Benefits and Impact:

The implementation of the EMPLOYEE MEDICAL Bill Tracking Portal brings numerous benefits to organizations. By automating and digitizing bill tracking processes, the portal reduces manual errors, minimizes processing times, and enhances overall operational efficiency. The transparent and user-friendly interfaces promote user satisfaction and engagement, while the robust security measures ensure the confidentiality and integrity of sensitive data. Ultimately, the portal contributes to improved accountability, streamlined workflows, and better decision-making.

Conclusion:

The EMPLOYEE MEDICAL Bill Tracking Portal is poised to revolutionize the way organizations manage their bill tracking processes. With its dual interfaces, comprehensive features, and user-centric design, the portal is set to enhance efficiency, transparency, and user satisfaction. By embracing this innovative solution, organizations can achieve greater accountability and operational excellence in their bill management endeavours.

