**Development**

**of**

**Gold Lab System Management System**

**for**

**CSC 387**

**Development of Gold Lab System Management System**

**A System Analysis and Design course Project Report**

**Submitted by -**

**Quad Coder**

|  |  |  |
| --- | --- | --- |
| **Sl No.** | **ID** | **Name** |
| **01** | **21303134** | **Md. Alamin** |
| 2 | 22103056 | Afifa Thashin |
| 3 | 22103382 | Nishpa Purification |

**Submitted To -**

Md. Saidur Rahman,

Assistant Professor

A logo with a black background

Description automatically generatedA logo with white letters and a green plus

Description automatically generated

**Department of Computer Science and Engineering**

College of Engineering and Technology

IUBAT- International University of Business Agriculture and Technology

**Letter of Transmittal**

**January 19, 2025**

**Md. Saidur Rahman**  
Assistant Professor  
Department of Computer Science and Engineering  
International University of Business Agriculture and Technology

**Subject: Submission of Project Report on “Gold Lab System Management”**

Dear Sir,

We are pleased to submit our project report titled **“Gold Lab System Management System”**, prepared as a part of the requirements for the **System Analysis and Design (CSC 387)** course. This report reflects our understanding, research, and efforts toward designing an efficient management system for a gold lab.

This project has been a valuable learning experience, and we have applied the theoretical concepts taught in class to address practical challenges in system analysis and design. We sincerely appreciate your guidance and support throughout the course, which helped us complete this project successfully.

The report is the collective effort of our group, **QuadCore Analysts**, consisting of the following members:

|  |  |  |
| --- | --- | --- |
| **Sl** | **ID** | **Name** |
| 1 | 21303134 | Md. Alamin |
| 2 | 22103056 | Afifa Thashin |
| 3 | 22103382 | Nishpa Purification |

We hope this report meets your expectations and provides valuable insights into our understanding of system design. We are open to any suggestions or feedback that can enhance our learning further.

Thank you for your guidance and encouragement.

Sincerely,

**QuadCore Analysts (Team Leder)**  
Md. Alamin (21303134)

# Abstract

The Gold Lab System Management is a web-based application designed to optimize and automate the processes of gold testing and hallmarking services. Currently, the company uses a manual system involving notebooks to record customer and jewelry shop details, item information, daily logs, and financial data. This manual approach is prone to errors, time-consuming, and lacks efficiency.

The proposed system aims to store detailed records of clients, jewelry shops, and jewelry items tested or hallmarked. It automates invoice generation, tracks financial transactions, and allows clients to verify invoices online using voucher numbers and customer IDs. The system calculates total charges based on the item type, quantity, and discounts, with real-time updates on payment statuses (paid or due). Additionally, it provides comprehensive business statistics and summaries for daily, weekly, monthly, and yearly performance. Key features include logical and physical system design, efficient database management using Entity Relationship Diagrams (ERD) and Data Flow Diagrams (DFD), and financial tracking for salaries, utility bills, and maintenance costs. The system offers scalability, reliability, and operational efficiency while ensuring user-friendly access for clients and business administrators.

By transitioning to this automated system, the company will reduce manual workload, minimize errors, and improve decision-making through detailed analytics and reporting, paving the way for enhanced customer satisfaction and business growth.

# Acknowledgment

We, the members of QuadCore Analysts, would like to express our heartfelt gratitude to everyone who supported us in the successful completion of this group report on the Gold Lab Management System.

First and foremost, we extend our sincere appreciation to our respected Assistant Professor, Md. Saidur Rahman, for his invaluable guidance, constructive feedback, and unwavering support throughout the project. His expertise and encouragement were instrumental in shaping this report and enhancing our understanding of the subject matter.

We also acknowledge the support and resources provided by our institution, IUBAT (International University of Business Agriculture and Technology), which created an enabling environment for us to complete this project efficiently.

A special thanks to all the members of QuadCore Analysts for their hard work, dedication, and excellent teamwork. Each member’s unique contributions played a vital role in the successful completion of this report.

Finally, we extend our gratitude to our families, friends, and peers for their continuous motivation and encouragement, which helped us stay focused and determined throughout this journey.

Thank you to everyone who contributed, directly or indirectly, to making this project a success.

QuadCore Analysts

BCSE Department, IUBAT

19-01-2025

# Table of Contents

Contents

[Abstract 4](#_Toc188185277)

[Acknowledgment 5](#_Toc188185278)

[Table of Contents 6](#_Toc188185279)

[Table Of Figure 9](#_Toc188185280)

[1. Introduction 1](#_Toc188185281)

[1.1. Introduction of Project 1](#_Toc188185282)

[1.2. Aim of Project 1](#_Toc188185283)

[1.3. System Study & Analysis 2](#_Toc188185284)

[1.3.1. System Analysis 2](#_Toc188185285)

[2. Existing System 2](#_Toc188185286)

[2.1. Existing Manual System 2](#_Toc188185287)

[2.2. Process of Existing System 2](#_Toc188185288)

[2.3. Problems with Existing System 2](#_Toc188185289)

[3. Proposed System 5](#_Toc188185290)

[3.1. Aim of the Proposed System 5](#_Toc188185291)

[3.1.1. Advantages of the Proposed System 5](#_Toc188185292)

[3.2. System Feasibility Study 7](#_Toc188185293)

[3.2.1. Technical Feasibility 7](#_Toc188185294)

[3.2.2. Economic Feasibility 8](#_Toc188185295)

[3.2.3. Operational Feasibility 8](#_Toc188185296)

[4. Proposed System Design 10](#_Toc188185297)

[4.1. Introduction of Proposed System 10](#_Toc188185298)

[4.1.1. Logical Design 10](#_Toc188185299)

[4.1.2. Physical Design 11](#_Toc188185300)

[4.1.3. Design/Specification Activities 11](#_Toc188185301)

[5. Implementation of Model 13](#_Toc188185302)

[5.1. Analysis Modeling & Design Methodologies 13](#_Toc188185303)

[5.1.1. Database Design 13](#_Toc188185304)

[5.1.2. Entity Relationship Model 14](#_Toc188185305)

[5.1.3. Identifying Entities 14](#_Toc188185306)

[5.1.4. Entity Relationship Diagram 15](#_Toc188185307)

[5.1.5. Relationship Cardinality 16](#_Toc188185308)

[5.1.6. Entity Relationship Diagram (Database Table Structure) 16](#_Toc188185309)

[5.2. System Description 17](#_Toc188185310)

[Data Flow Diagram (DFD) Levels for Gold Lab Management System 17](#_Toc188185311)

[DFD Level 0: Context Diagram 17](#_Toc188185312)

[DFD Level 1: Decomposing the System into Sub-processes 18](#_Toc188185313)

[DFD Level 2: Detailed Processes 20](#_Toc188185314)

[1. Process: Manage Customers 20](#_Toc188185315)

[2. Process: Manage Invoices 20](#_Toc188185316)

[3. Process: Manage Expenses 21](#_Toc188185317)

[4. Process: Generate Reports 21](#_Toc188185318)

[5. Process: Manage Jewelry Shops 21](#_Toc188185319)

[6. Process: Revenue Summary 21](#_Toc188185320)

[5.2.4. Effort Distribution 22](#_Toc188185321)

[5.2.5. Task Distribution 22](#_Toc188185322)

[5.2.6. Time Chart for Activities 23](#_Toc188185323)

[5.2.7. System Specification 23](#_Toc188185324)

[5.2.8. Project Cost Estimation 23](#_Toc188185325)

[5.2.8.2. Software costs 23](#_Toc188185326)

[5.2.8.3. Other Costs 24](#_Toc188185327)

[5.3. System Testing 24](#_Toc188185328)

[5.3.1 System Testing 24](#_Toc188185329)

[5.3.2. Test Plan 25](#_Toc188185330)

[5.3.3. Types of Testing Performed 25](#_Toc188185331)

[5.3.4. Key Testing Outcomes 26](#_Toc188185332)

[6. System Requirements 27](#_Toc188185333)

[6.1. System Requirements 27](#_Toc188185334)

[6.1.1. Hardware Requirements 27](#_Toc188185335)

[6.1.2. Software Requirements 28](#_Toc188185336)

[6.1.3. Screenshots for Project Key Features 30](#_Toc188185337)

[7. Conclusion & Upcoming Features: 45](#_Toc188185338)

[7.1 Conclusion 45](#_Toc188185339)

[7.2. Upcoming Features 45](#_Toc188185340)

[References 47](#_Toc188185341)

[Bibliography 48](#_Toc188185342)

[Appendices 49](#_Toc188185343)

# Table Of Figure

[Figure 1 - Entiity Relationship Diagram 15](https://d.docs.live.net/6827155863af1bee/System%20Analysis%20and%20Design%20SAD%20course%20Project%20Report%20CBT%20Gold%20Lab%20(1).docx#_Toc188185241)

[Figure 2 - ERD of Database Table Structure 17](https://d.docs.live.net/6827155863af1bee/System%20Analysis%20and%20Design%20SAD%20course%20Project%20Report%20CBT%20Gold%20Lab%20(1).docx#_Toc188185242)

[Figure 3 - Level 0 DFD or Context Diagram 18](https://d.docs.live.net/6827155863af1bee/System%20Analysis%20and%20Design%20SAD%20course%20Project%20Report%20CBT%20Gold%20Lab%20(1).docx#_Toc188185243)

[Figure 4 - Level 1 DFD 20](https://d.docs.live.net/6827155863af1bee/System%20Analysis%20and%20Design%20SAD%20course%20Project%20Report%20CBT%20Gold%20Lab%20(1).docx#_Toc188185244)

[Figure 5 - Login Page 30](#_Toc188185245)

[Figure 6 - Screenshort of dashboard. 30](#_Toc188185246)

[Figure 7 - Screenshort of home page 31](#_Toc188185247)

[Figure 8 - Customer List 31](#_Toc188185248)

[Figure 9 - Jwelry Item List 32](#_Toc188185249)

[Figure 10 - Jewerlry Shop List 32](#_Toc188185250)

[Figure 11 - Creating Invoice for Customer 33](#_Toc188185251)

[Figure 12 - Invoice Details 34](#_Toc188185252)

[Figure 13 - All Invoice Record 35](#_Toc188185253)

[Figure 14 - Send Invoice via Email 35](#_Toc188185254)

[Figure 15 - Download / Print Invoice 36](#_Toc188185255)

[Figure 16 - Downloaded or Printable Invoice Format 36](#_Toc188185256)

[Figure 17 - Verify Invoice Over online 37](#_Toc188185257)

[Figure 18 - Matching Invoice over online 37](#_Toc188185258)

[Figure 19 - Daily Summary Report 38](#_Toc188185259)

[Figure 20 - Monthly Summary Report 38](#_Toc188185260)

[Figure 21 - Custom Filtering Based Summary Report 39](#_Toc188185261)

[Figure 22 - Find Customer's (invoice) History 39](#_Toc188185262)

[Figure 23 - Expense Category CRUD 40](#_Toc188185263)

[Figure 24 - Expense List 40](#_Toc188185264)

[Figure 25 - Expense Summary Filtered Based 41](#_Toc188185265)

[Figure 26 - Expense Monthly Summary by Year wise 42](#_Toc188185266)

[Figure 27 – Current Month Revenue Summary 43](#_Toc188185267)

[Figure 28 - Filter Based Revenue Summary 44](#_Toc188185268)

# 

# 1. Introduction

## 1.1. Introduction of Project

The Gold Lab Management System is a comprehensive software solution designed to optimize and modernize the operations of jewelry businesses. It aims to simplify and automate critical daily tasks such as generating invoices, managing customer records, tracking expenses, and creating detailed financial reports. By replacing traditional manual record-keeping methods with an efficient digital system, this software ensures greater accuracy, speed, and convenience in managing business operations.

Built using advanced technologies, the system leverages Spring Boot for robust backend processing, MySQL for reliable database management, and Thymeleaf combined with Bootstrap to deliver an intuitive and responsive user interface. Key features include real-time data visualization, customized reporting options, and seamless management of revenue and expenses.

In the context of countries like Bangladesh, where manual processes are still prevalent in many small and medium-sized enterprises, the Gold Lab Management System offers a transformative approach to jewelry shop management. By embracing digital solutions, business owners can save valuable time, enhance accuracy, and focus on scaling their operations effectively while maintaining transparency and control.

## 1.2. Aim of Project

The aim of this project is to simplify and automate the management process of jewelry shops in Bangladesh. By providing a complete digital solution, the system ensures:

1.Faster and more accurate invoice generation.

2.Simplified customer and jewelry shop management.

3.Detailed expense and income tracking.

4.Generation of professional reports (daily, monthly, yearly).

5.Clear insights into financial performance through revenue summaries.

The ultimate goal is to provide a reliable tool for jewelry businesses to replace their existing manual systems and enhance their efficiency and profitability.

## 1.3. System Study & Analysis

### 1.3.1. System Analysis

Jewelry shops that rely on manual processes face numerous challenges in their day-to-day operations. Manual methods for managing invoices, expenses, and reports are often time-consuming, error-prone, and inefficient. These traditional practices can result in data inaccuracies, misplaced records, and increased operational overhead.

A detailed analysis of the manual system revealed the following key issues:

• Manual invoices are prone to errors and may not be stored properly. • Tracking expenses and income requires additional effort and calculations.

• Business owners lack the tools to understand financial trends easily. • Generating reports for daily or monthly activities is tedious.

To overcome these challenges, the Gold Lab Management System was designed as a digital solution. It streamlines jewelry shop operations by automating tasks such as record-keeping, invoicing, and financial reporting. The system ensures accuracy, reduces time spent on repetitive tasks, and offers real-time insights into business performance. By implementing this software, jewelry shop owners can focus more on growing their business and less on managing operational complexities.

# 2. Existing System

## 2.1. Existing Manual System

In Bangladesh, many jewelry shops continue to operate using traditional manual systems for managing daily business activities. These systems involve maintaining physical records of customers, invoices, jewelry items, and expenses. Business owners use registers or notebooks to store information, and invoices are generated using pre-printed receipt books. Financial tracking is done by hand, often using calculators to determine revenues and expenses.

While this system may seem manageable for small-scale businesses, it becomes inefficient as the business grows, leading to complications in data management, accuracy, and scalability.

## 2.2. Process of Existing System

The current process of running a jewelry shop using a manual system includes:

1.**Recording Customer Details**   
Customer information, such as names, contact numbers, and purchase history, is written in registers or notebooks. Updating or retrieving these records is often time-consuming.

2.**Generating Invoices**   
Invoices are handwritten on receipt books. The calculations are done manually, which increases the likelihood of errors.

3.**Tracking Expenses and Revenue**   
Expenses and income are noted down separately, and owners manually calculate total expenses, income, and revenue. This process lacks accuracy and real-time insights.

4.**Storing Records**   
Physical files or folders are used to store all data, including customer records, invoices, and financial details. Over time, these records pile up, making retrieval and organization challenging.

5.**Generating Reports**   
Reports, such as monthly revenue or yearly expenses, are manually compiled by going through individual records, which is tedious and time-consuming.

## 2.3. Problems with Existing System

The manual system presents several challenges for jewelry shop owners:

1.**High Likelihood of Errors**   
Manual calculations and handwritten records often lead to inaccuracies, especially when handling large volumes of data.

2.**Difficulty in Data Retrieval and Management**   
Finding specific information, such as a customer’s purchase history or expense details from previous months, becomes challenging in a physical record-keeping system.

3.**No Real-Time Insights or Summaries**   
Business owners cannot instantly view their revenue, top-selling items, or expense breakdown. This lack of insights hinders decision-making and financial planning.

4.**Inefficient Report Generation**   
Creating reports manually requires significant effort and time, delaying critical financial assessments and impacting business efficiency.

5.**Data Loss Risks**   
Physical records are prone to loss or damage due to mishandling, natural disasters, or deterioration over time, risking important business information.

# 3. Proposed System

## 3.1. Aim of the Proposed System

The **Gold Lab Management System** is designed to automate and centralize various operations for jewelry shop management. The aim is to provide shop owners with tools to streamline their day-to-day processes, such as managing customer records, tracking revenues, and generating insightful reports. By addressing the inefficiencies of manual operations, the proposed system ensures a seamless workflow, minimal errors, and maximum productivity.

The specific objectives of the system are:

• **Automation of Processes**: Eliminate repetitive manual tasks like invoice generation and expense calculations by introducing automated workflows.

• **Centralized Data Management**: Provide a single platform where all business-related data— customers, invoices, expenses, and revenues—can be securely stored and accessed.

• **Real-Time Insights**: Offer updated insights into the financial health of the business, such as current revenues, top-performing product categories, and profit margins.

• **Scalability**: Ensure the system is capable of supporting the growing needs of the jewelry business, whether it’s handling more customers or managing multiple shop locations.

### 3.1.1. Advantages of the Proposed System

#### 1. Real-Time Data Processing and Insights

The system processes all data immediately upon entry, allowing shop owners to:

* Access live updates on revenue, expenses, and profits without waiting for manual calculations.
* Monitor financial trends as they happen, enabling better decision-making.
* Identify profitable products or services quickly based on live sales data.

#### **2. Accurate Tracking of Expenses, Revenues, and Profits**

Manual systems are prone to calculation errors, especially when dealing with large datasets. The **Gold Lab Management System** ensures:

• **Expense Tracking**: All expenses are categorized and stored in the database, making it easy to review spending patterns.

• **Revenue Monitoring**: Invoices are recorded automatically, providing an accurate picture of total income.

• **Profit Calculations**: The system calculates profits by subtracting total expenses from revenue,

giving a clear view of business performance.

#### 3. Automatic Invoice Generation and Verification

Creating invoices manually can be time-consuming and error-prone. This system automates the

process by:

* Using predefined templates to generate professional-looking invoices quickly.
* Cross-referencing invoices with customer and jewelry shop data to ensure accuracy.
* Storing all invoices securely for future reference.

#### 4. Detailed and Customizable Reports

Reports are a vital tool for analyzing business performance. The system provides:

• **Daily, Monthly, and Yearly Summaries**: Generate comprehensive reports on revenues, expenses,

and profits for any selected time frame.

• **Filtering Options**: Customize reports by filtering data based on categories, customers, or dates.

• **Visual Representations**: Include charts and graphs in reports for better understanding of

financial trends.

#### **5. Ease of Data Retrieval**

Searching through physical records is time-consuming and unreliable. With the proposed system:

• All data is stored in a centralized database, allowing for instant retrieval of specific records.

• Advanced search functionality helps users locate invoices, customer histories, or expense details

within seconds.

#### **6. Improved Security and Data Backup**

The proposed system offers better data security compared to manual methods. It includes:

• **Role-Based Access Control**: Ensure only authorized personnel can access sensitive data.

• **Regular Backups**: Protect data from accidental loss or corruption by implementing scheduled backups.

• **Encryption**: Safeguard sensitive information like customer details and financial data through encryption.

## 3.2. System Feasibility Study

When developing a system like the **Gold Lab Management System**, it’s crucial to ensure that it is feasible in terms of technical, economic, and operational aspects. Below is a detailed analysis of the feasibility of this proposed system.

### 3.2.1. Technical Feasibility

This aspect evaluates whether the technology required for the system is available, reliable, and suitable for implementation.

• **Modern Tools and Technologies**:   
 o **Spring Boot Framework**: This Java-based framework simplifies application development by providing built-in features for database integration, security, and performance optimization. It is well-suited for enterprise-level applications like this one.

o **MySQL Database**: A relational database management system, MySQL is known for its robustness, scalability, and ability to handle large volumes of data. It ensures that all records (customers, invoices, expenses) are securely stored and easily retrievable.

o **Bootstrap 5.3.3**: This responsive frontend framework ensures the system’s user interface is visually appealing and accessible across devices, including desktops, tablets, and smartphones.

o **Data Security:**

Role-based access control (RBAC) is implemented to restrict unauthorized access to

sensitive data.

In conclusion, the technical tools and frameworks ensure the system’s reliability, security, and ability to handle the current and future needs of a jewelry shop.

### 3.2.2. Economic Feasibility

This aspect determines whether the cost of developing and implementing the system is justified by the benefits it brings.

• **Development Costs**:   
 o The use of open-source technologies like Spring Boot, MySQL, and Bootstrap minimizes

software licensing costs.

|  |  |  |
| --- | --- | --- |
| • | o | The system’s development relies on widely available resources, reducing training and |
| hiring expenses.  **Operational Savings**: | |
| o | Automation of tasks like invoice generation, report creation, and revenue calculation |

reduces the need for manual labor, saving on staffing costs.

|  |  |  |
| --- | --- | --- |
| • | o Improved accuracy in financial calculations minimizes errors, avoiding potential losses.  **Return on Investment (ROI)**: | |
| o | By providing real-time insights and improving decision-making, the system enables |

better financial management, leading to higher profitability.

o The time saved in managing day-to-day operations allows shop owners to focus on

strategic activities like customer satisfaction and business growth.

The proposed system provides significant economic value, offering long-term benefits that far outweigh the initial investment.

### 3.2.3. Operational Feasibility

Operational feasibility examines whether the system can function effectively in the jewelry shop environment and whether it is user-friendly for shop owners and staff.

• **Ease of Use**:   
 o The system features a clean and intuitive interface, reducing the learning curve for new users.

|  |  |  |
| --- | --- | --- |
| • | o | Guided workflows for tasks like customer management, invoice creation, and report |
| generation ensure even non-technical users can operate the system efficiently. **Improved Efficiency**: | |
| o | Automating repetitive tasks like manual bookkeeping and report generation reduces |

human effort and ensures consistent performance.

|  |  |  |
| --- | --- | --- |
| •  • | o Real-time data processing allows users to make informed decisions quickly.  **Adaptability**: | |
| o | The system is designed to accommodate varying business needs, such as managing |
| multiple shops, tracking different expense categories, and generating custom reports. **Support and Maintenance**: | |
| o | Regular updates ensure the system stays compatible with evolving technologies and |

business requirements.

|  |  |  |
| --- | --- | --- |
| • | o | Comprehensive documentation and user guides are provided for troubleshooting and |
| operational support.  **Resilience**: | |

o Automatic database backups protect against data loss, ensuring business continuity in case of hardware failures or other disruptions.

In summary, the **Gold Lab Management System** is designed to be operationally effective, improving efficiency and accuracy while being easy to adopt by the jewelry shop staff.

By ensuring technical, economic, and operational feasibility, the proposed system is well-positioned to meet the needs of jewelry shop owners while providing a robust platform for growth and innovation.

# 4. Proposed System Design

The design of the **Gold Lab Management System** focuses on creating a robust, efficient, and user-friendly application. It ensures seamless integration of various components, enabling accurate data management and reporting for the jewelry shop.

## 4.1. Introduction of Proposed System

The proposed system introduces a structured design approach, integrating logical and physical components to achieve the desired functionality. It is built with a focus on scalability, security, and responsiveness, meeting the requirements of modern-day business operations.

### 4.1.1. Logical Design

The logical design outlines the system’s structure, ensuring a clear understanding of how data flows between components and entities.

• **Core Entities**:   
 o **Customers**: Stores details such as name, contact information, and associated jewelry shop.

o **Invoices**: Tracks transactions, including item details, total price, discount, and payment status.

o **Expenses**: Categorizes expenses by type, amount, and description, with associated dates.

|  |  |  |
| --- | --- | --- |
| • | o | **Jewelry Shops**: Manages shop-specific data for tracking revenue and expenses. |
| o | **Items**: Tracks jewelry item details like weight, carat, and price. |
| o **Revenue**: Summarize the income and expenses  **Relationships**: | |
| • | o | Customers and invoices are connected through a one-to-many relationship. | |
| o | Expenses are categorized to enable detailed analysis and summaries. | |
| o | Jewelry shops are linked to invoices and customers to ensure consistent revenue | |
| tracking.  **Workflow**: | | |
| o | User inputs trigger workflows like invoice creation, expense logging, and report | |

generation.

o Validation mechanisms ensure data integrity, reducing errors during data entry.

This logical design ensures data consistency and simplifies data retrieval, making the system

intuitive for users.

### 4.1.2. Physical Design

The physical design focuses on the system’s technical implementation, including database

structures, application logic, and user interface.

|  |  |  |
| --- | --- | --- |
| •  • | **Database**: | |
| o | The backend database uses **MySQL**, a reliable and scalable RDBMS. |
| o | Tables are normalized to reduce redundancy and maintain data integrity. |
| o | Relationships are implemented using foreign keys to link entities like invoices, |
| customers, and expenses.  **Application Logic**: | |
| o | **Spring Boot** is the core framework for server-side logic, ensuring modular, testable, and |

maintainable code.

|  |  |  |
| --- | --- | --- |
| •  • | o RESTful APIs are used for interactions between the frontend and backend.  **Frontend**: | |
| o | **Thymeleaf** is used for rendering dynamic HTML pages. |
| o | **Bootstrap 5.3.3** ensures the frontend is responsive, accessible, and visually appealing. |
| o Chart.js provides interactive visualizations for expense and revenue summaries.  **Deployment**: | |
| o | The system can be deployed on any server with a MySQL database installed, with the |

capability to create the database automatically if it doesn’t exist.

### 4.1.3. Design/Specification Activities

To achieve the functionality of the system, the following design specifications are implemented:

• **Entities**:

o **Customers**: Includes fields for name, contact information, and due amounts.

o **Invoices**: Includes details like invoice date, total price, discount, paid amount, and due

amount.

o **Expenses**: Includes fields for category, amount, description, and date.

|  |  |  |
| --- | --- | --- |
| • | o | **Jewelry Shops**: Includes shop details for associating customers and invoices. |
| o **Items**: Includes details like item type, weight, carat, and associated price.  **Reports**: | |
| o | **Monthly and Yearly Summaries**: Provide aggregated data for income, expenses, and |

revenue.

|  |  |  |
| --- | --- | --- |
| • | o | **Expense Categories**: Show detailed breakdowns by category. |
| o **Revenue Details**: Highlight profit and loss trends over selected periods.  **User Interaction**: | |
| o | The dashboard provides a summary of key metrics, including revenue, total expenses, and |

customer details.

o Filtering options allow users to view data for specific months, years, or categories.

o Export features enable saving reports as PDF or CSV for offline analysis.

This design ensures that the **Gold Lab Management System** meets business requirements while remaining user-friendly, scalable, and efficient.

# 5. Implementation of Model

The implementation of model section will cover the systematic implementation of the **Gold Lab Management System** by detailing the model, design methodologies, database structure, and other aspects crucial to its development and functionality.

## 5.1. Analysis Modeling & Design Methodologies

This subsection mainly focuses on how the system was analyzed and designed. It outlines the database structure, entity relationships, and methodologies adopted for implementation.

### 5.1.1. Database Design

The database is the backbone of the **Gold Lab Management System**, storing and managing all the data for customers, invoices, expenses, jewelry shops, and revenue summaries.

• **Tables**:

o **Customer**:

▪ Stores customer details such as full name, contact information, and due amounts.

o **Invoice**:

▪ Tracks all invoices, including date, total price, paid amount, and due amount.

o **Expense**:

▪ Logs all expenses, categorized by type, date, and description.

* **Jewelry Shop**:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| • | ▪ | | | Contains details about the shops associated with the invoices and customers. |
| o | **Revenue Summary**: | | |
| **Features**: | | ▪ | Stores aggregated revenue, expenses, and profit/loss data for reporting. |

**Features:**

* The database follows 3rd Normal Form(3NF) to eliminate redundancy and ensure data integrity

o Indexes are created for frequently queried fields such as customer\_id, invoice\_id,

and date for faster data retrieval.

|  |  |  |
| --- | --- | --- |
| • | o **Foreign keys** enforce relationships and ensure referential integrity between tables.  **Automatic Database Creation**: | |
| o | The system is configured to create the required database automatically using Spring |

Boot’s spring.jpa.hibernate.ddl-auto=update setting.

### 5.1.2. Entity Relationship Model

An entity relationship diagram (ERD), also known as an entity relationship model, is a graphical representation that depicts relationships among people, objects, places, concepts or events in an information technology (IT) system. In this project, relationships are implemented using Spring Boot’s **JPA annotations**:

|  |  |  |
| --- | --- | --- |
| •  • | **Annotations Used**: | |
| o | @OneToMany: For linking one customer to multiple invoices. |
| o | @ManyToOne: For associating each invoice with a single customer and jewelry shop. |
| o @JoinColumn: Specifies foreign keys in relationships.  **Example**: | |

@Entity   
public class Invoice {   
 @Id   
 @GeneratedValue(strategy = GenerationType.IDENTITY) private Long id;

@ManyToOne   
 @JoinColumn(name = "customer\_id", nullable = false) private Customer customer;

@ManyToOne   
 @JoinColumn(name = "jewelry\_shop\_id", nullable = false) private JewelryShop jewelryShop;   
}

This ensures that relationships are strongly defined, enabling smooth data retrieval and manipulation.

### 5.1.3. Identifying Entities

The system identifies the following key entities based on functional requirements:

1.**Customer**:

|  |  |
| --- | --- |
| o | Attributes: id, fullName, phoneNumber, dueAmount. |
| o Relationships: One-to-many with invoices.  2.**Invoice**: | |
| o | Attributes: id, invoiceDate, totalPrice, paid, due. |
| o Relationships: Many-to-one with customers and jewelry shops.  3.**Expense**: | |
| o | Attributes: id, date, category, amount, description. |
| o Relationships: None (category is a standalone entity).  4.**JewelryShop**: | |
| o | Attributes: id, name, location. |
| o  5.**Item**: | Relationships: One-to-many with invoices. |
| o | Attributes: id, type, weight, price. |
| o | Relationships: Many-to-one with invoices. |

### 5.1.4. Entity Relationship Diagram

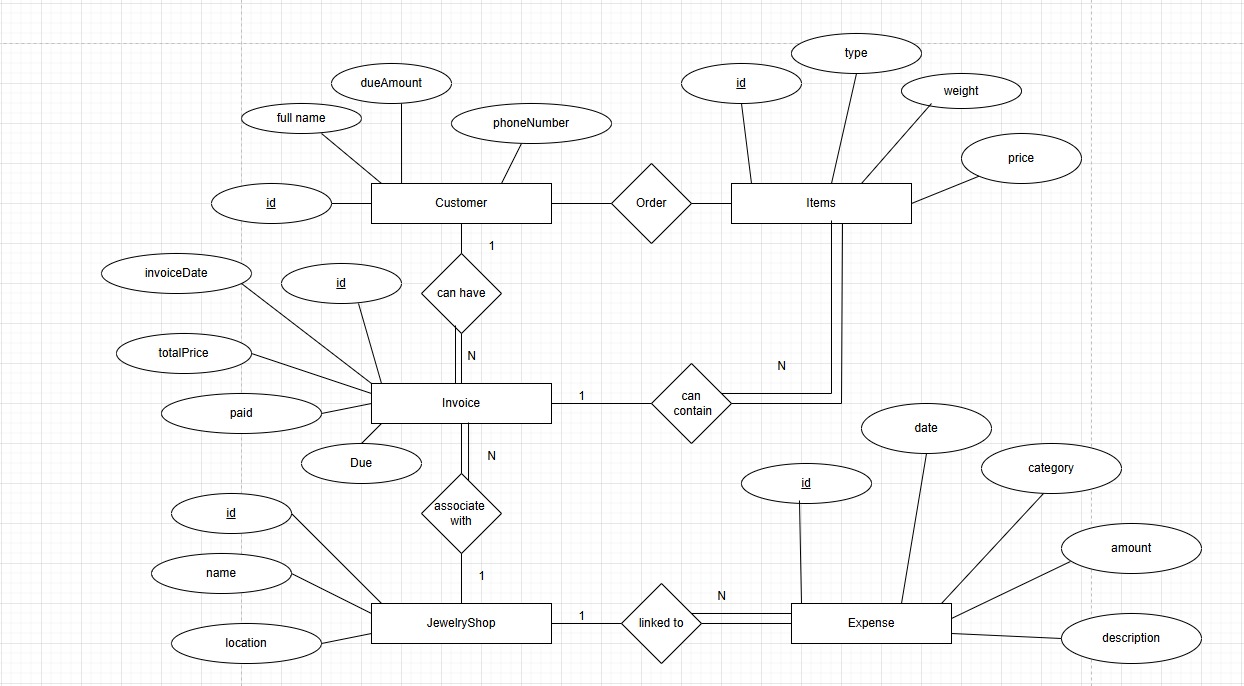
The **Entity Relationship Diagram (ERD)** visually represents the relationships between entities. It highlights how data flows within the system and how entities are interconnected.

Figure 1 - Entiity Relationship Diagram

### 5.1.5. Relationship Cardinality

Cardinality defines the number of instances of one entity related to another.

**Defined Relationships**:

* **Customer and Invoice**: (One-to-Many) A single customer can have multiple invoices.
* **Invoice and JewelryShop**: (Many-to-One) Each invoice is associated with one jewelry shop.
* **Expense and Category**: (Many-to-One) Each expense is associated with one category.
* **JewelryShop and Expense: (**One to many) Each jewelry shop can record multiple expenses, but each expense belongs to one shop.

### 5.1.6. Entity Relationship Diagram (Database Table Structure)

The **ERD for Database Table Structure** is provided as a visual aid (refer to the appendices). It includes all entities and their relationships, as defined in the logical design. **MySQL Workbench** were used for diagram generation. Each table includes:

* **Primary Key:** A unique value that can be used to refer to an entire record. Like Id.
* **Foreign Key**:  A column in a table that references a primary key in another table.
* **Fields:**  An attribute of a record in a database table.

A screenshot of a computer

Description automatically generated

Figure 2 - ERD of Database Table Structure

## 5.2. System Description

System description will outline how data flows within the **Gold Lab Management System**, along with the methodologies used for designing and implementing the system. It also covers effort distribution, task management, and cost estimation, ensuring an accurate depiction of the project's development lifecycle.

## Data Flow Diagram (DFD) Levels for Gold Lab Management System

Below are the steps for creating DFD levels for the **Gold Lab Management System**, based on our project discussions. A DFD typically consists of **external entities**, **processes**, **data stores**, and **data flows**.

DFD Level 0: Context Diagram

The context diagram provides a high-level overview of the system. It shows the system as a single process and its interaction with external entities.

**Components:**

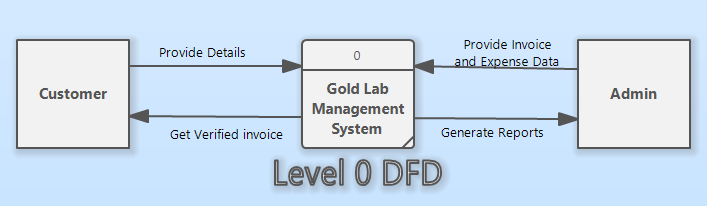
* **External Entities**:
  + Customer
  + Admin
* **Single Process**:
  + Gold Lab Management System
* **Data Flows**:
  + Customer provides details (e.g., name, contact) → System
  + Admin provides inputs (e.g., expense data, invoice data) → System
  + System generates reports (e.g., revenue summary, invoices) → Admin
  + System sends invoices or confirmations → Customer

Figure 3 - Level 0 DFD or Context Diagram

### DFD Level 1: Decomposing the System into Sub-processes

In **DFD Level 1**, we break down the system into its major functional components. Each component interacts with the data flows and entities identified in the context diagram (Level 0).

#### Sub-processes in DFD Level 1:

1. **Customer Management**:
   * Input: Customer details.
   * Process: Add, update, or retrieve customer records.
   * Output: Customer data is stored or fetched for other operations like invoice generation.
2. **Invoice Management**:
   * Input: Customer selection, jewelry item details, jewelry shop, and transaction details.
   * Process: Generate invoices and calculate revenue and due amounts.
   * Output: Invoice details stored and displayed for verification or reporting.
3. **Expense Management**:
   * Input: Expense category, description, amount, and date.
   * Process: Add, update, and categorize expenses.
   * Output: Expense records for monthly/yearly analysis.
4. **Report Generation**:
   * Input: Data from invoices, expenses, and customer records.
   * Process: Generate detailed reports for revenue, expenses, and profits (daily, monthly, yearly).
   * Output: Customizable reports viewable or exportable (CSV/PDF).
5. **Admin Operations**:
   * Input: Admin credentials and operational commands.
   * Process: Manage system roles, notifications, and logs.
   * Output: Admin privileges applied, and operations logged.
6. **Revenue Calculation**:
   * Input: Data from invoices and expenses.
   * Process: Calculate total income, total expenses, and net revenue.
   * Output: Revenue summaries and analytics.

#### Data Flows Between Components:

* **Customer Management** provides customer data to **Invoice Management**.
* **Invoice Management** sends invoice details to **Report Generation** and **Revenue Calculation**.
* **Expense Management** contributes data to **Report Generation** and **Revenue Calculation**.
* **Admin Operations** overves security, logs, and system settings.
* **Report Generation** uses processed data from multiple sources for summaries.

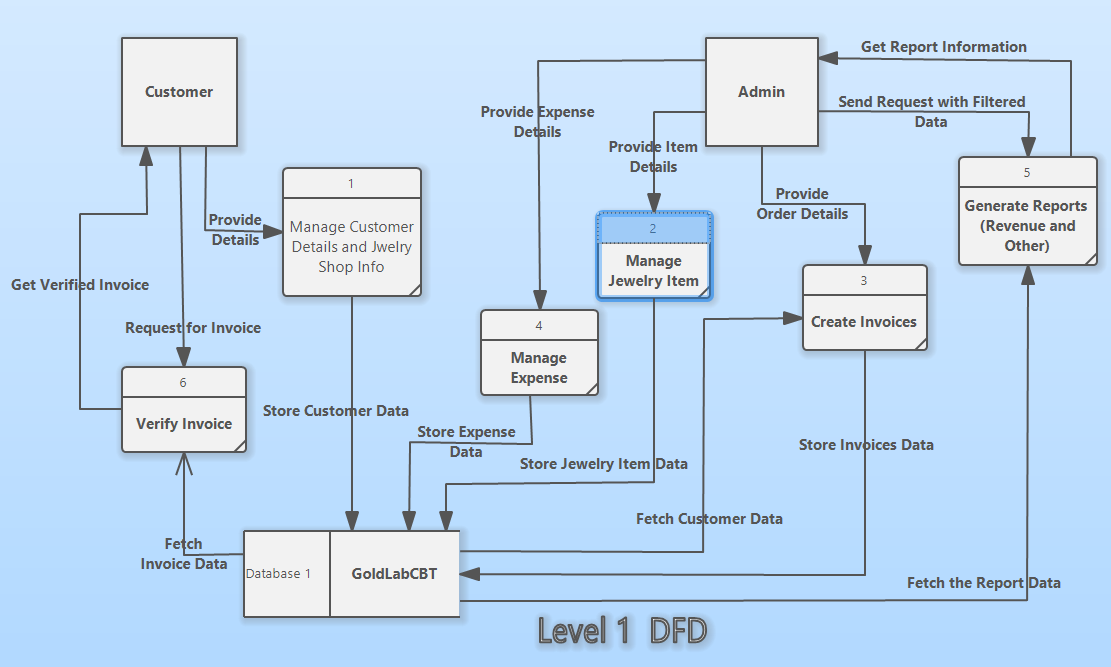


Figure 4 - Level 1 DFD

DFD Level 2: Detailed Processes

Each high-level process is broken down into detailed sub-processes.

**1. Manage Customers:**

* **1.1. Add Customer**: Save customer details to **Customer Data**.
* **1.2. Update Customer**: Modify customer details in **Customer Data**.
* **1.3. Delete Customer**: Remove customer from **Customer Data**.

**2. Create Invoices:**

* **2.1. Select Customer**: Fetch customer data from **Customer Data**.
* **2.2. Select Jewelry Item**: Fetch item data from **Jewelry Data**.
* **2.3. Calculate Total**: Compute total cost.
* **2.4. Save Invoice**: Store invoice in **Invoice Data**.

**3. Track Expenses:**

* **3.1. Select Expense Category**: Fetch categories from **Expense Data**.
* **3.2. Add Expense**: Record expense details.
* **3.3. Update Expense**: Modify an existing expense.

**4. Generate Reports:**

* **4.1. Revenue Report**:
  + Calculate income from **Invoice Data**.
  + Deduct expenses from **Expense Data**.
* **4.2. Expense Report**:
  + Summarize expenses by category from **Expense Data**.
* **4.3. Monthly Summary**:
  + Fetch monthly data from **Invoice Data** and **Expense Data**.

### 1. ****Process: Manage Customers****

* **1.1** Add Customer: Input customer details (name, contact, address, etc.) into the system.
* **1.2** Update Customer: Modify existing customer details.
* **1.3** View Customer Details: Retrieve and display customer information.
* **1.4** Delete Customer: Remove a customer record from the database.

### 2. ****Process: Manage Invoices****

* **2.1** Create Invoice: Generate a new invoice by selecting customer, jewelry shop, and items.
* **2.2** Update Invoice: Edit existing invoice details (e.g., date, paid status).
* **2.3** View Invoice: Display invoice details for review or printing.
* **2.4** Verify Invoice: Validate invoice authenticity using a unique voucher number.

### 3. ****Process: Manage Expenses****

* **3.1** Add Expense: Log a new expense with category, date, and amount.
* **3.2** Update Expense: Modify details of an existing expense entry.
* **3.3** View Expense Report: Generate expense summaries by date, category, or month.
* **3.4** Delete Expense: Remove an expense entry.

### 4. ****Process: Generate Reports****

* **4.1** Daily Reports: Summarize daily income, expenses, and revenue.
* **4.2** Weekly Reports: Provide a breakdown of weekly performance.
* **4.3** Monthly Reports: Highlight income, expenses, and revenue for a specific month.
* **4.4** Yearly Reports: Summarize yearly financial data.
* **4.5** Export Reports: Generate and download reports in CSV or PDF format.

### 5. ****Process: Manage Jewelry Shops****

* **5.1** Add Jewelry Shop: Register a new jewelry shop.
* **5.2** Update Jewelry Shop: Edit existing shop details.
* **5.3** View Shops: List all registered shops.
* **5.4** Delete Jewelry Shop: Remove a shop from the database.

### 6. ****Process: Revenue Summary****

* **6.1** Calculate Total Expenses: Summarize all expenses for a given period.
* **6.2** Calculate Total Income: Summarize total income from invoices for the same period.
* **6.3** Calculate Revenue: Deduct total expenses from total income to get net revenue.

### 5.2.4. Effort Distribution

Effort was distributed across several phases, ensuring a balanced approach to development:

|  |  |  |
| --- | --- | --- |
| **Phase** | **Effort (Hours)** | **Percentage** |
| Requirement Analysis | 15 | 11% |
| System Design | 25 | 18% |
| Development | 82 | 61% |
| Testing | 6 | 5% |
| Deployment and Training | 6 | 5% |

### 5.2.5. Task Distribution

Tasks were divided among the team members based on expertise:

|  |  |  |
| --- | --- | --- |
| **Task** | **Assigned To** | **Description** |
| Database, ERD | Afifa, Nishpa | Designing and normalizing tables. |
| Backend and Frontend Development | Md.Alamin | Writing Business Logic for CRUD operations and integrated with frontend. |
| Frontend  Design | Afifa, Nishpa | Designing responsive interfaces using Bootstrap |
| Testing | Md. Alamin,Affa,Nishpa | Performing unit, integration, and system tests. |
| Documentation and Reports | Alamin, Afifa, Nishpa | Preparing user manuals and project reports |

### 5.2.6. Time Chart for Activities

The project timeline was mapped using a **Gantt Chart**, ensuring timely completion of

milestones:

|  |  |  |  |
| --- | --- | --- | --- |
| Activity | Starting Date | Ending Date | Duration |
| Requirement Analysis | 26-11-2024 | 01-12-2024 | 5 Days |
| System Design | 12-12-2024 | 22-12-2024 | 10 Days |
| Development Phase | 27-12-2024 | 17-01-2025 | 20 Days |
| Testing | 18-01-2025 | 18-01-2025 | 7 Days |
| Deployment | 19-01-2025 | 19-01-2025 | 3 Days |

### 5.2.7. System Specification

The system specification outlines the technical details for deployment and operation:

1.**Hardware Requirements**:

|  |  |
| --- | --- |
| o | Minimum RAM: 4 GB. |
| o | Processor: Dual-core 2.0 GHz or higher. |
| o | Storage: 10 GB free space. |
| o Network: Stable internet connection for API access.  2.**Software Requirements**: | |
| o | Backend: Java 17, Spring Boot. |
| o | Frontend: Thymeleaf, Bootstrap 5.3.3. |
| o | Database: MySQL 8.0. |

### 5.2.8. Project Cost Estimation

The cost estimation includes hardware, software, and additional costs:

5.2.8.1. Hardware Costs

|  |  |  |  |
| --- | --- | --- | --- |
| **Item** | **Quantity** | **Cost per Unit (BDT)** | **Total cost (BDT)** |
| Server Rent Monthly | 1 | 500 | 500 |
| Networking Equipment Monthly | 1 | 500 | 500 |

### 5.2.8.2. Software costs

|  |  |
| --- | --- |
| **Item** | **Cost (BDT)** |
| MySQL License (if needed) | Free |
| Development tools | Free |

### 5.2.8.3. Other Costs

|  |  |
| --- | --- |
| **Item** | **Cost (BDT)** |
| Training and Support | 300 |
| Miscellaneous | 100 |

**Total Estimated Cost**: BDT 1,400.

**5.2.9. Cost-Benefit Analysis**

The proposed system offers significant benefits that outweigh the costs:

1.**Cost Savings**:

|  |  |
| --- | --- |
| o | Reduces manual labor and associated costs. |
| o Minimizes errors, reducing financial losses.  2.**Efficiency Gains**: | |
| o | Automates repetitive tasks. |
| o Generates detailed reports instantly.  3.**Scalability**: | |
| o | The system can handle increased data volume as the business grows. |

## 5.3. System Testing

System testing ensures that the **Gold Lab Management System** functions as intended, meets

user requirements, and is free of critical bugs. It covers functionality, performance, security, and usability testing.

### 5.3.1 System Testing

The system underwent comprehensive testing at multiple stages to ensure its robustness and reliability. This process was designed to identify and resolve any defects prior to deployment, ensuring the application operates seamlessly in a production environment.

**Primary Objectives of System Testing:**

* Validate all core functionalities, including invoice generation, expense tracking, and revenue summarization.
* Verify compatibility across various browsers and devices.
* Ensure the system can manage concurrent user activity without experiencing performance issues.

### 5.3.2. Test Plan

The **Test Plan** includes detailed test cases covering all critical functionalities and scenarios. Below is an overview of the testing process:

1. **Test Cases for Invoice Generation**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Description** | **Expected Result** | **Actual** |
| **ID** | **Result** |
| TC001 | Generate an invoice for a customer. | Invoice generated with accurate total | Pass |
| price and due amount. |
| TC002 | Attempt to generate an invoice | System prompts user to fill mandatory | Pass |
| without required fields. | fields. |
| TC003 | Verify invoice date validations. | Invoice date cannot be in the future. | Pass |

### 5.3.3. Types of Testing Performed

1.**Unit Testing**:   
 o Focused on individual modules like invoice management, expense tracking, and report

generation.

o Tools used: JUnit and Mockito.

2.**Integration Testing**:   
 o Verified data flow between modules, such as linking invoices to customers and expenses.

* Ensured proper communication with the database and APIs.

3.**Performance Testing**:

o Tested the system's response time and ability to handle concurrent users.

o Tools used: Apache JMeter.

4.**Security Testing**:

o Verified role-based access controls to ensure data confidentiality.

o Tested protection against SQL injection and cross-site scripting (XSS).

5.**User Acceptance Testing (UAT)**:

o Conducted with a small group of end-users to validate the system's usability and

functionality.

o Ensured proper communication with the database and APIs.

### 5.3.4. Key Testing Outcomes

1.**Accuracy**:

o All calculations for invoices, expenses, and revenue summaries were validated to be

correct.

2.**Usability**:

o The system was intuitive and user-friendly, as confirmed during UAT.

3.**Reliability**:

o Stress testing showed the system could handle up to 200 concurrent users without

degradation.

# 6. System Requirements

The System Requirement outlines the system requirements for running the **Gold Lab Management System** and provides a step-by-step guide for setting up and using the application.

## 6.1. System Requirements

The system requirements are categorized into **hardware** and **software** specifications to ensure

optimal performance.

### 6.1.1. Hardware Requirements

1.**Processor**:

o Minimum: Intel Core i5 (6th generation or later).

o Recommended: Intel Core i7 or equivalent AMD processor for enhanced performance.

2.**RAM**:

o Minimum: 8 GB.

o Recommended: 16 GB or more for multitasking and running multiple modules

concurrently.

3.**Storage**:

|  |  |
| --- | --- |
| o | Minimum: 50 GB of free disk space. |
| o Recommended: SSD for faster database operations and application startup.  4.**Display**: | |
| o | Resolution: 1366x768 (minimum). |
| o | Recommended: Full HD (1920x1080) or higher for better UI clarity. |

### 

### 6.1.2. Software Requirements

1.**Operating System**:

|  |
| --- |
| o Compatible with Windows 10/11 or Linux distributions like Ubuntu 20.04+.  2.**Database**:   o MySQL 8.0 or later.  3.**Java Development Kit (JDK)**:   o Version: Java 17 or higher.  4.**Build Tool**:   o Apache Maven (version 3.8 or later).  5.**Web Browser**:   o Latest versions of Chrome, Firefox, or Edge for accessing the web interface.  6.**Spring Boot Dependencies**:   * Spring Boot 3.x or later. * Thymeleaf (For template rendering) |

**Step 1: Install Prerequisites**

1.Install **Java JDK 17+** and ensure it's configured in your system's environment variables.

bash

Copy code

java -version

2.Install **MySQL 8.0+** and create a root user with appropriate permissions.

**Step 2: Configure Application Properties**

1.Open the file src/main/resources/application.properties.

2.Update the database connection details:

spring.datasource.url=jdbc:mysql://localhost:3306/GoldLabCBT spring.datasource.username=root   
spring.datasource.password=yourpassword   
spring.jpa.hibernate.ddl-auto=update

**Step 3: Build and Run the Application**

1.Open the terminal or IDE (e.g., IntelliJ, Eclipse).

2.Navigate to the project directory and build the application:

mvn clean install

3.Start the Spring Boot application:

java -jar target/GoldLabCBT-0.0.1-SNAPSHOT.jar

4.Access the application via http://localhost:8080.

**Step 4: Login and Explore Features**

1.Use the default admin credentials to log in:

o Username: adminUsername

o Password: adminPassword

2.Start managing customers, invoices, expenses, and revenue summaries.

### 

### 6.1.3. Screenshots for Project Key Features

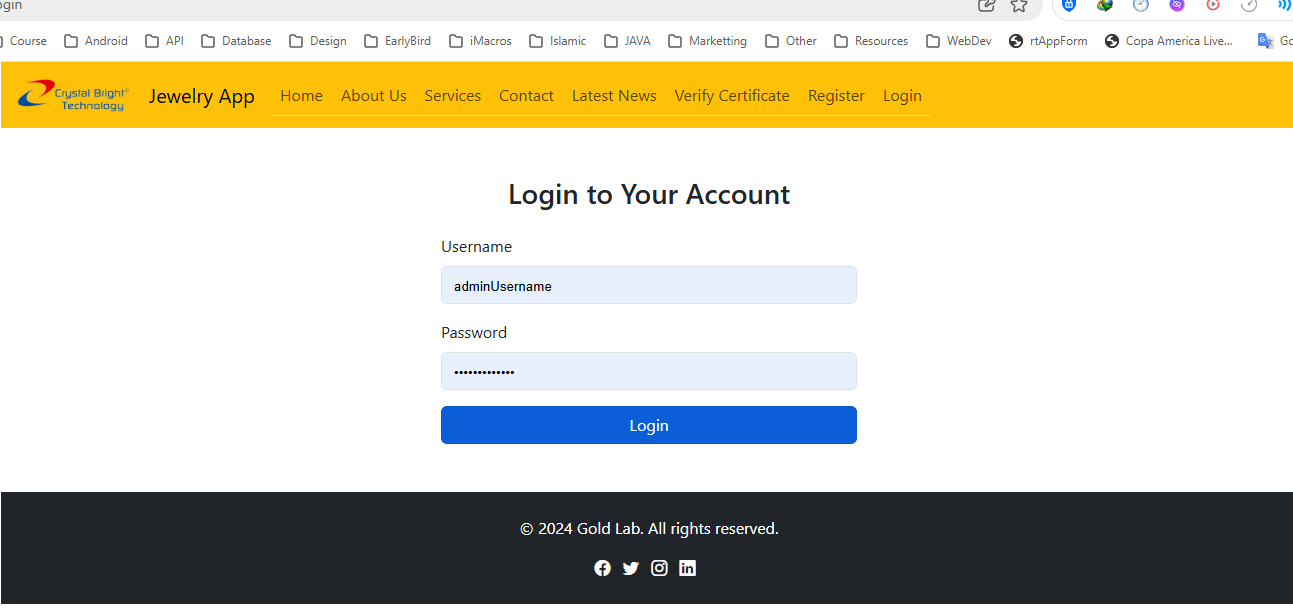


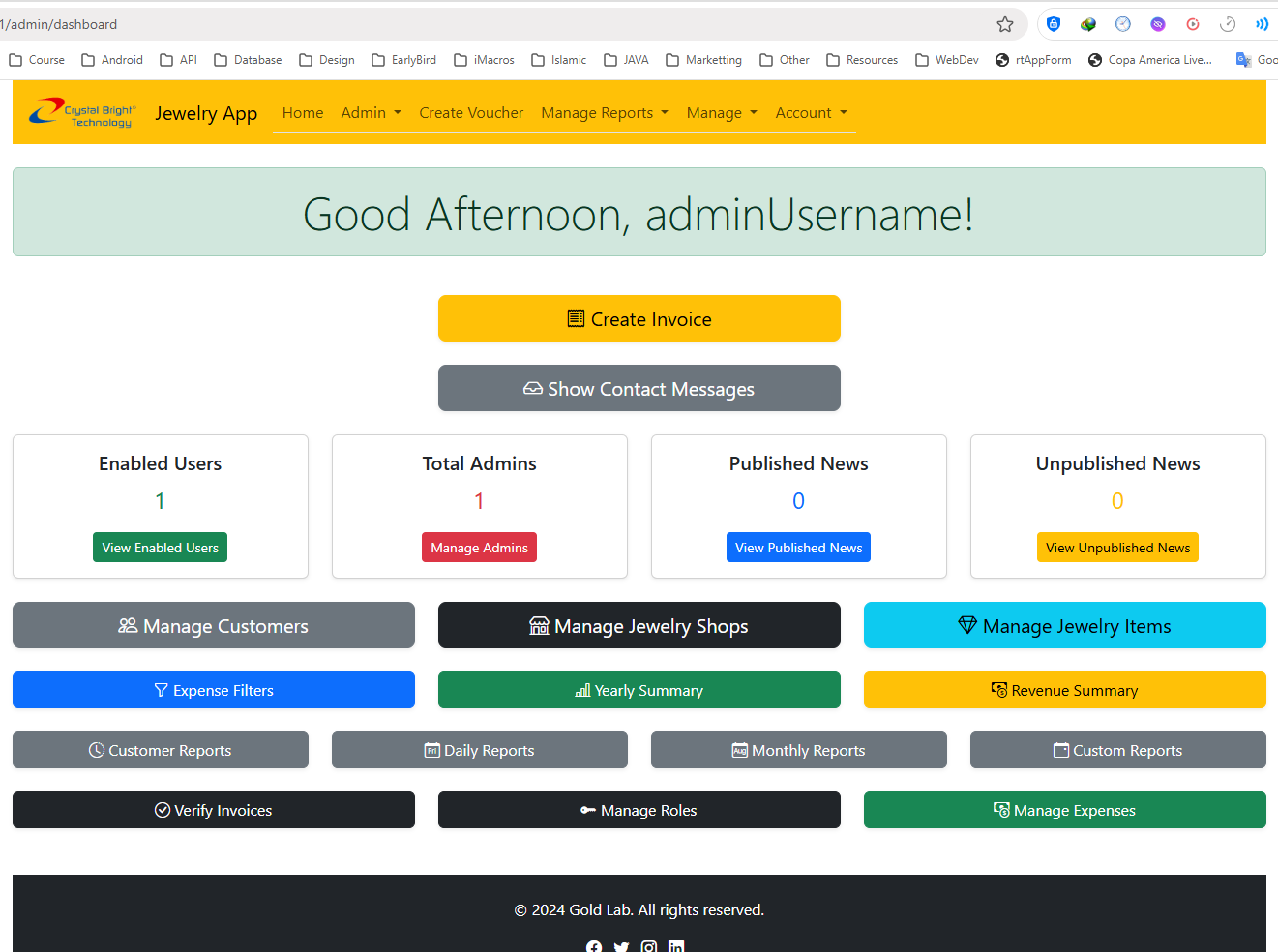
Figure 5 - Login Page

Figure 6 Screenshort of dashboard.

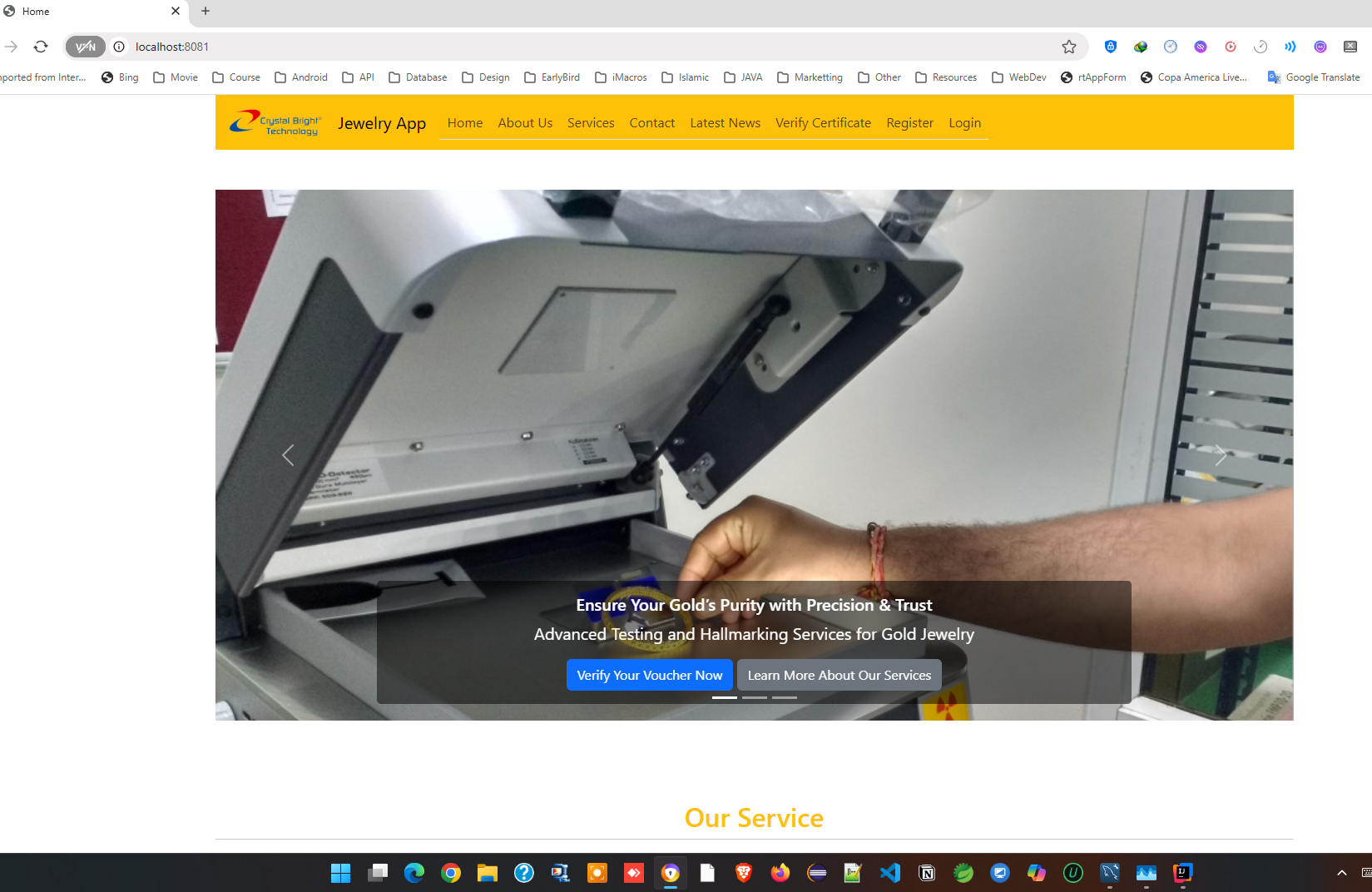


Figure 7 - Screenshort of home page

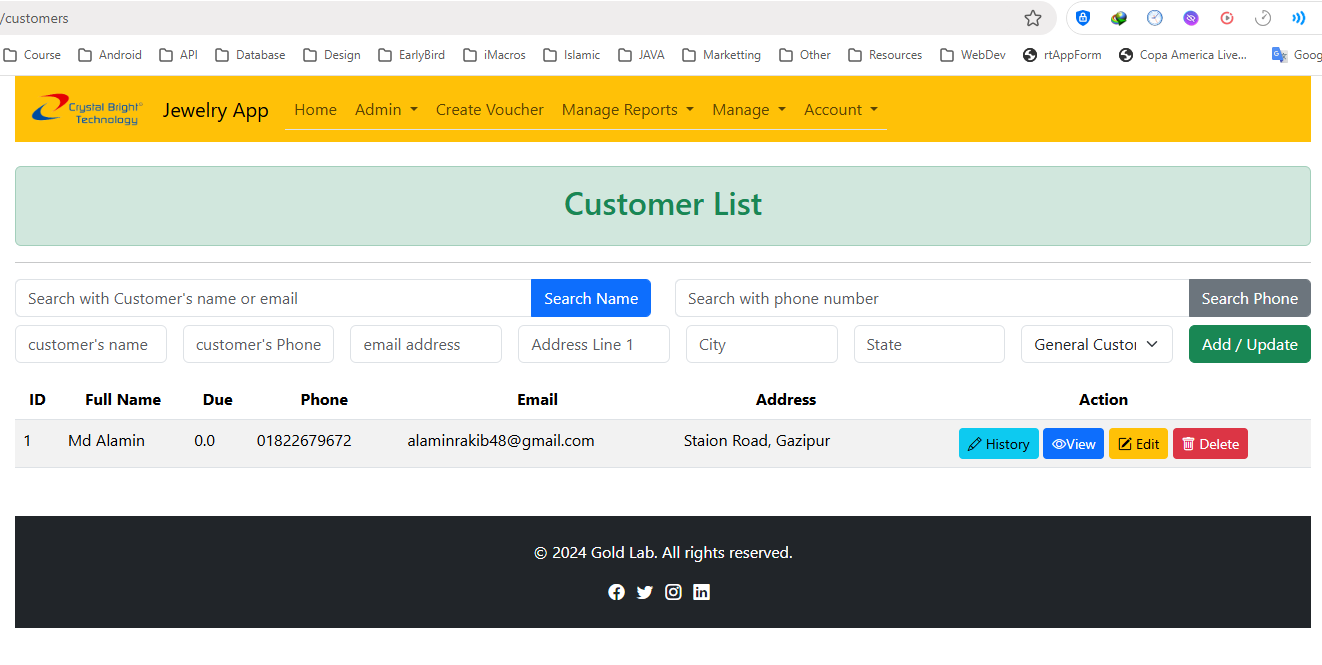
****

Figure 8 - Customer List

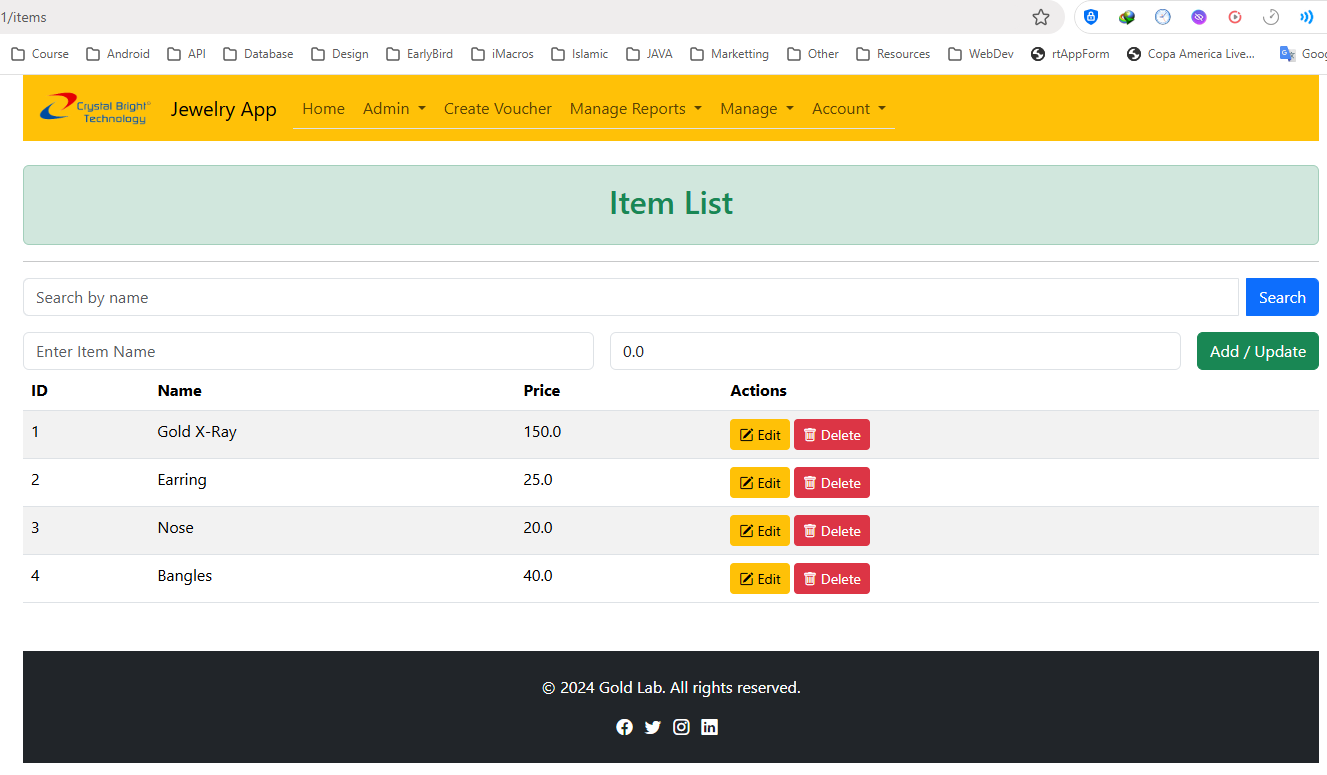


Figure 9 - Jwelry Item List

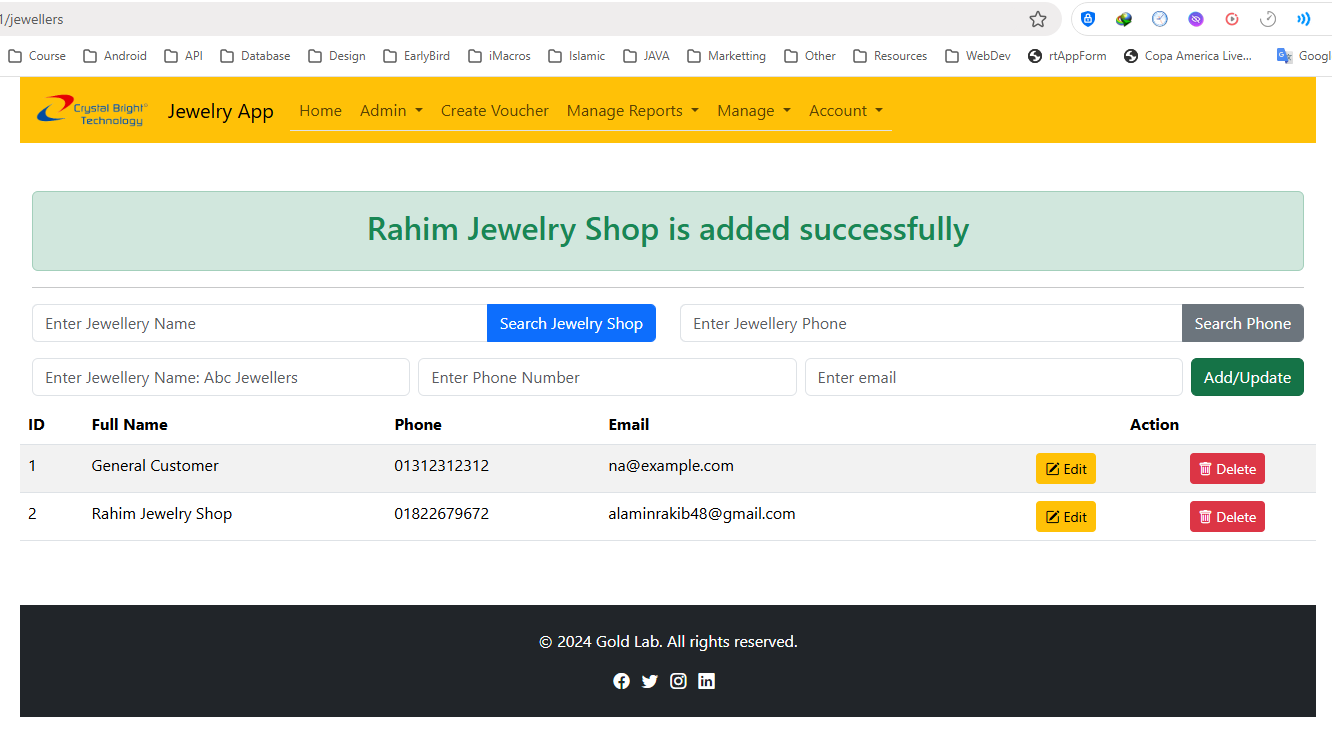


Figure 10 - Jewerlry Shop List

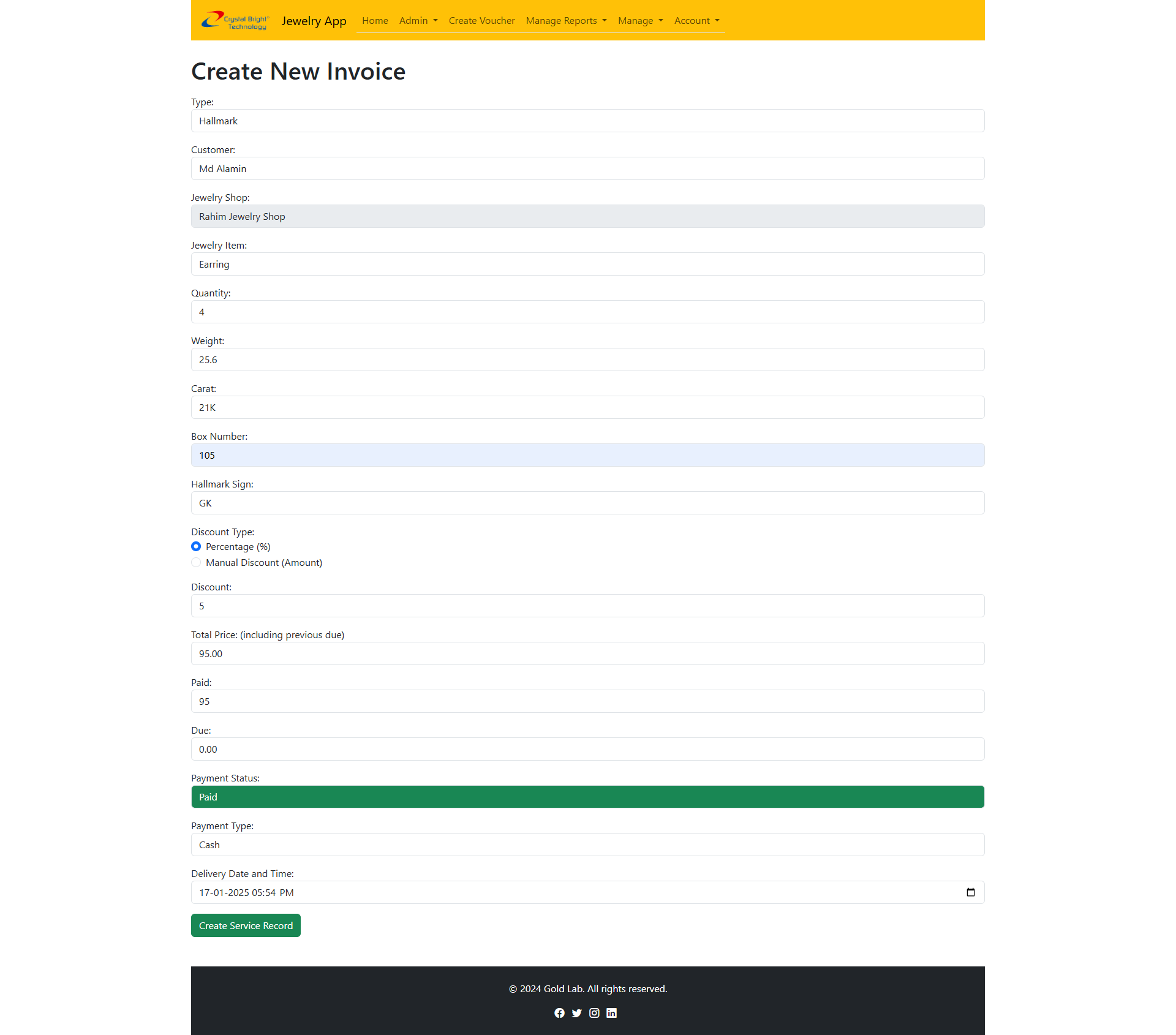


Figure 11 - Creating Invoice for Customer

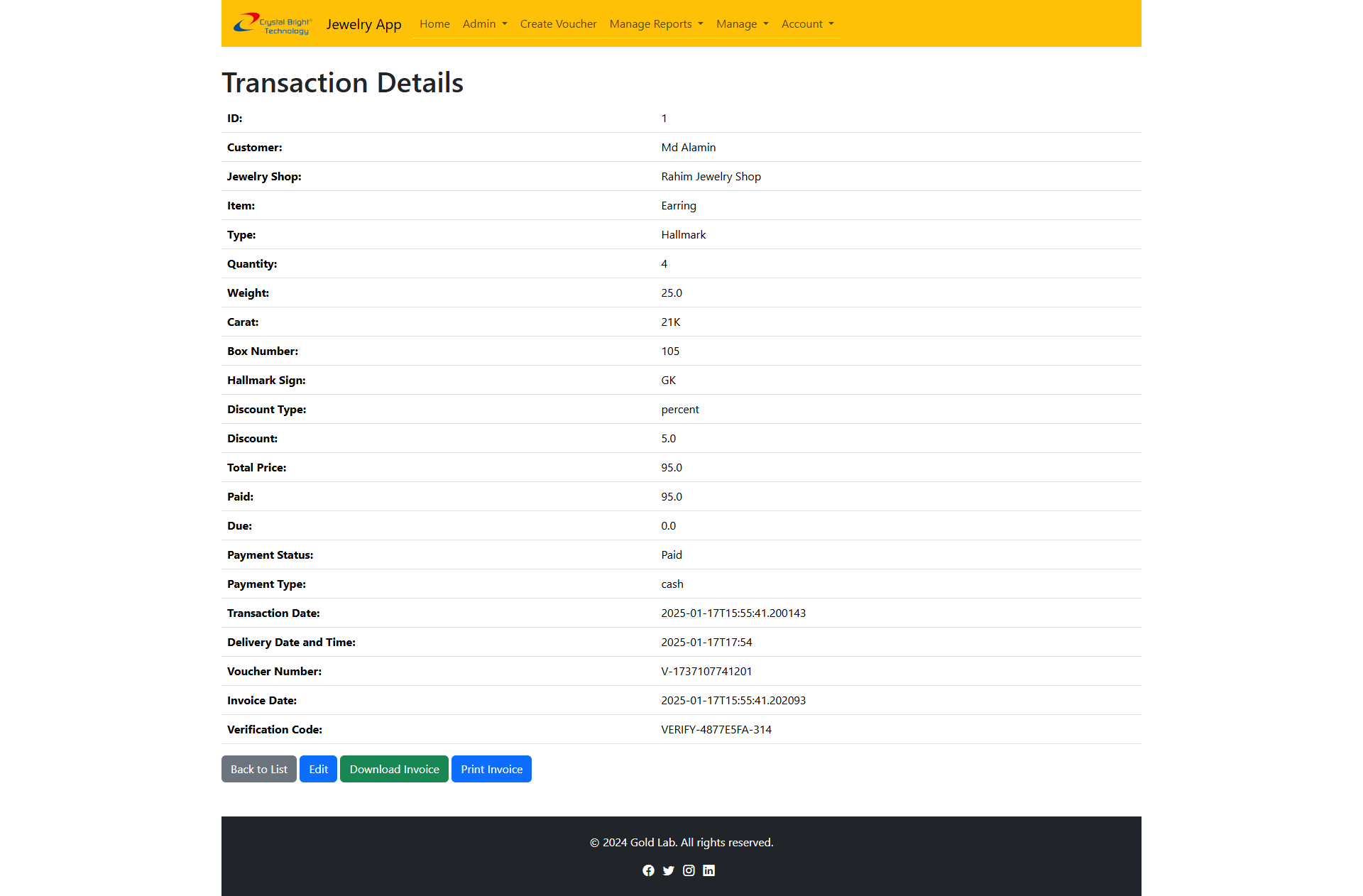


Figure 12 - Invoice Details

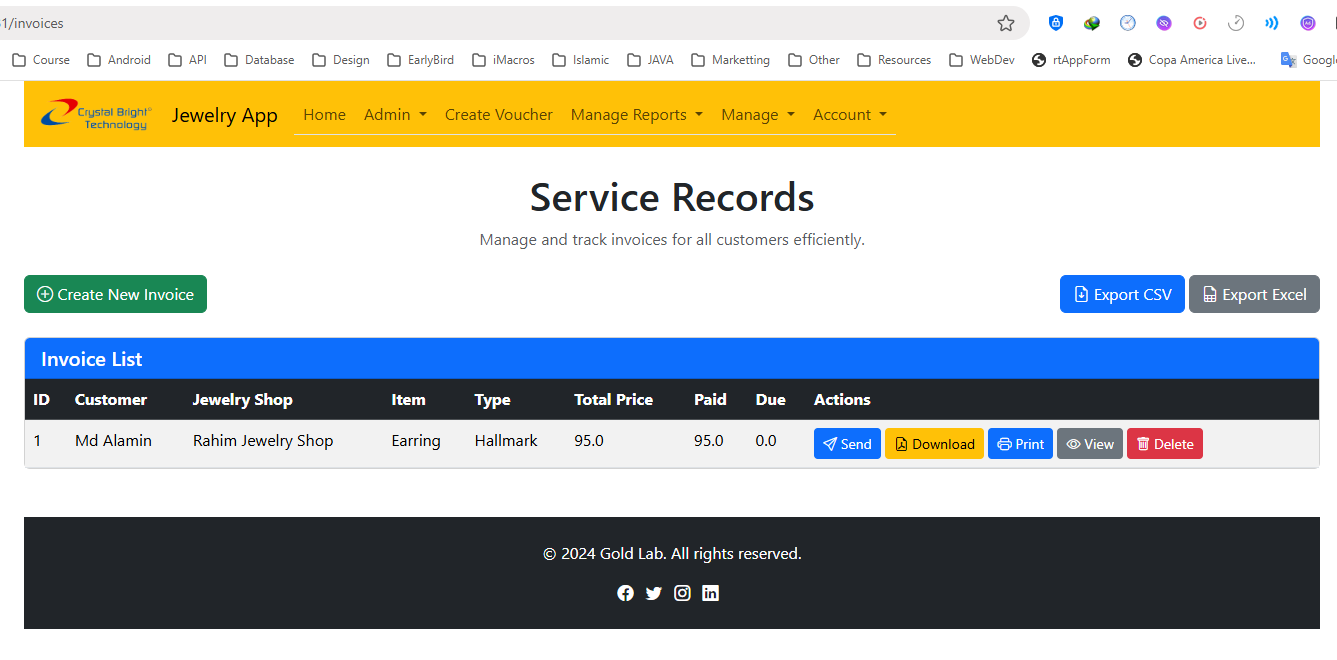


Figure 13 - All Invoice Record

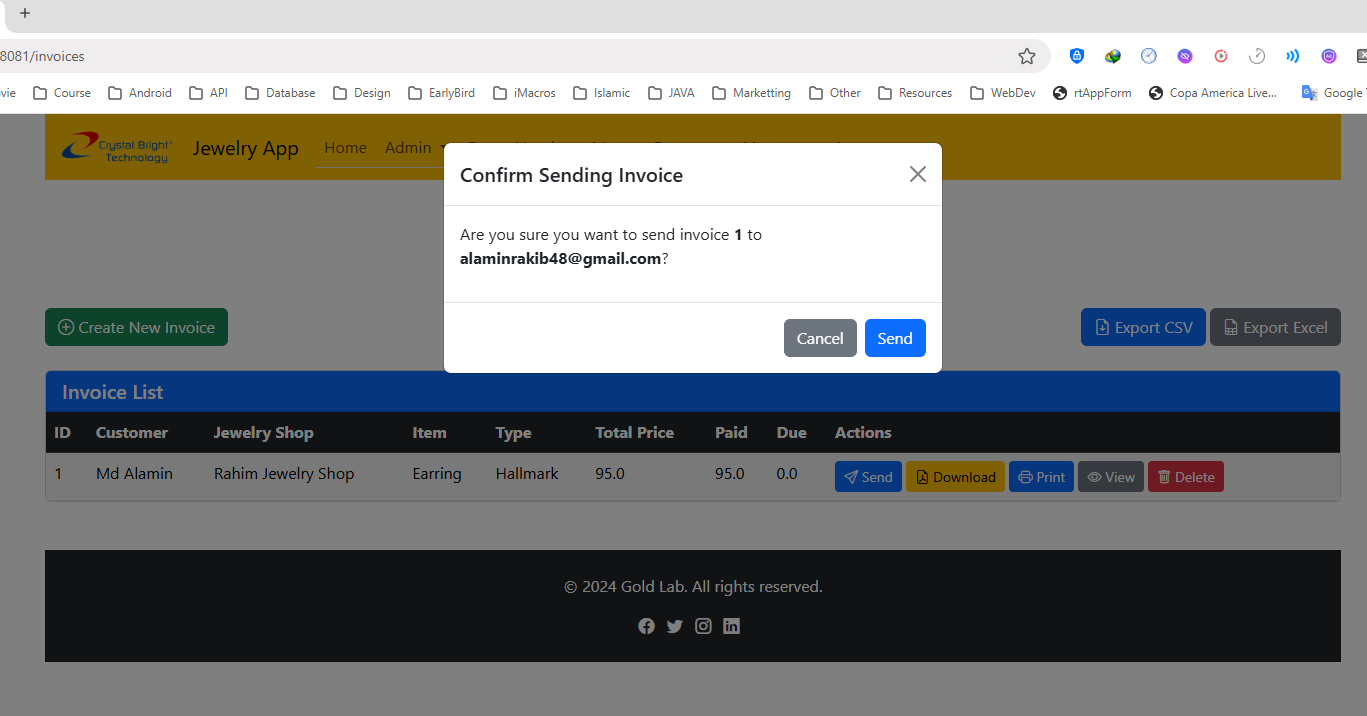


Figure 14 - Send Invoice via Email

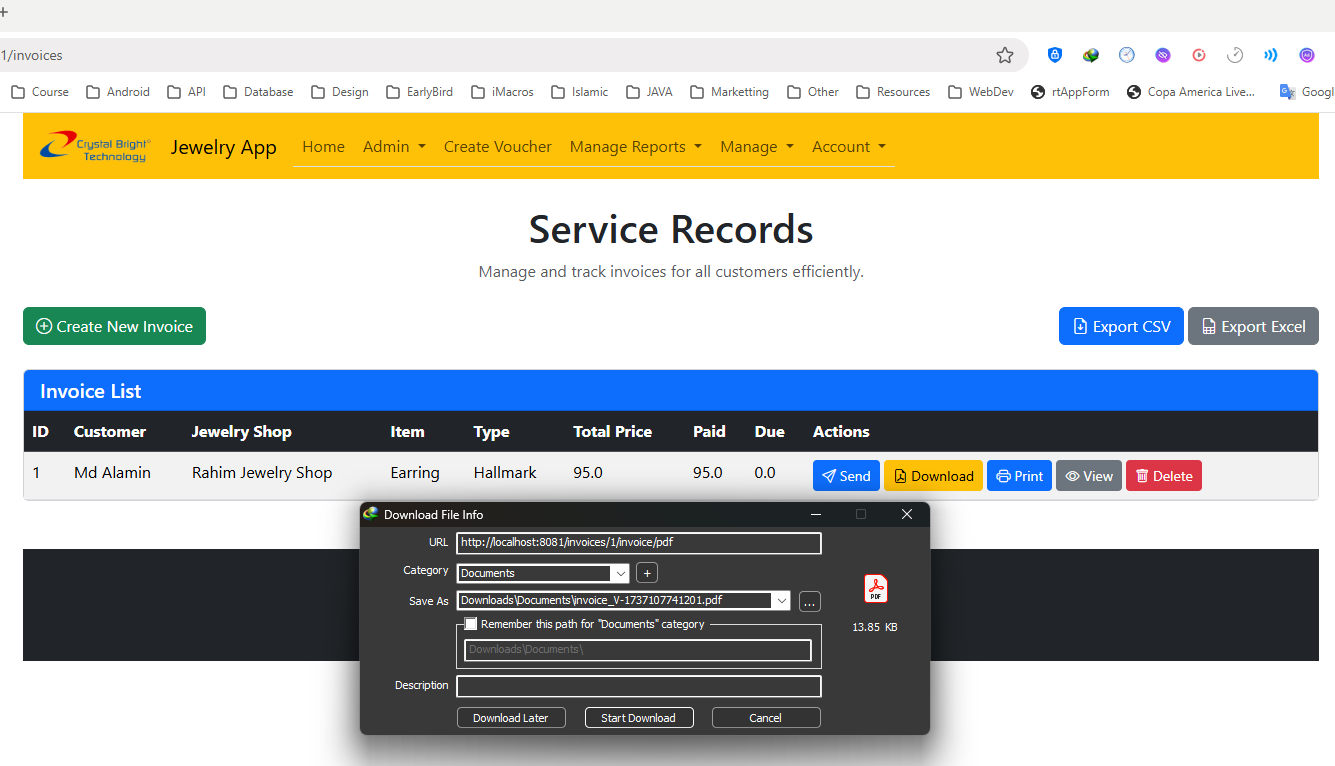


Figure 15 - Download / Print Invoice

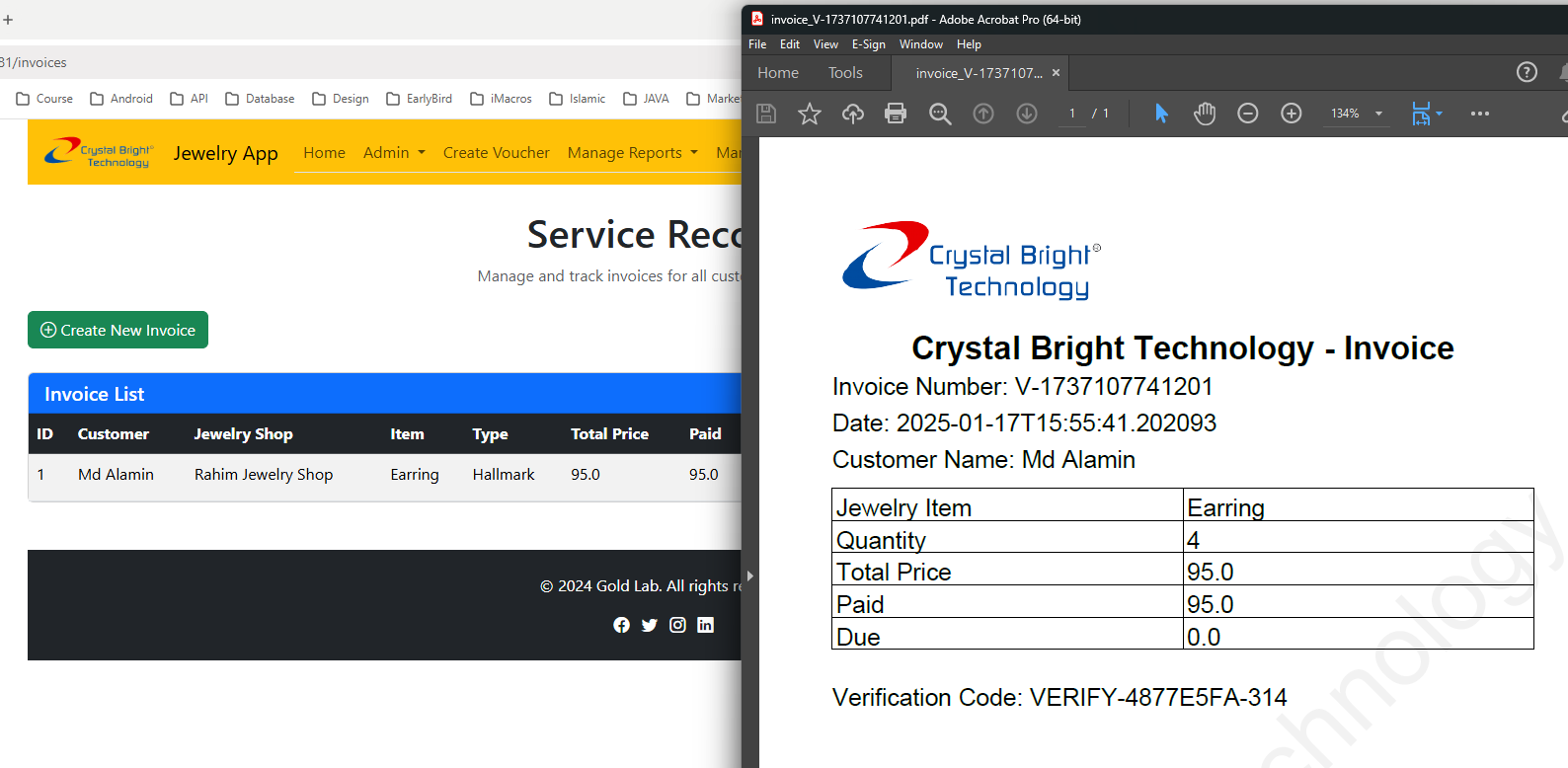


Figure 16 - Downloaded or Printable Invoice Format

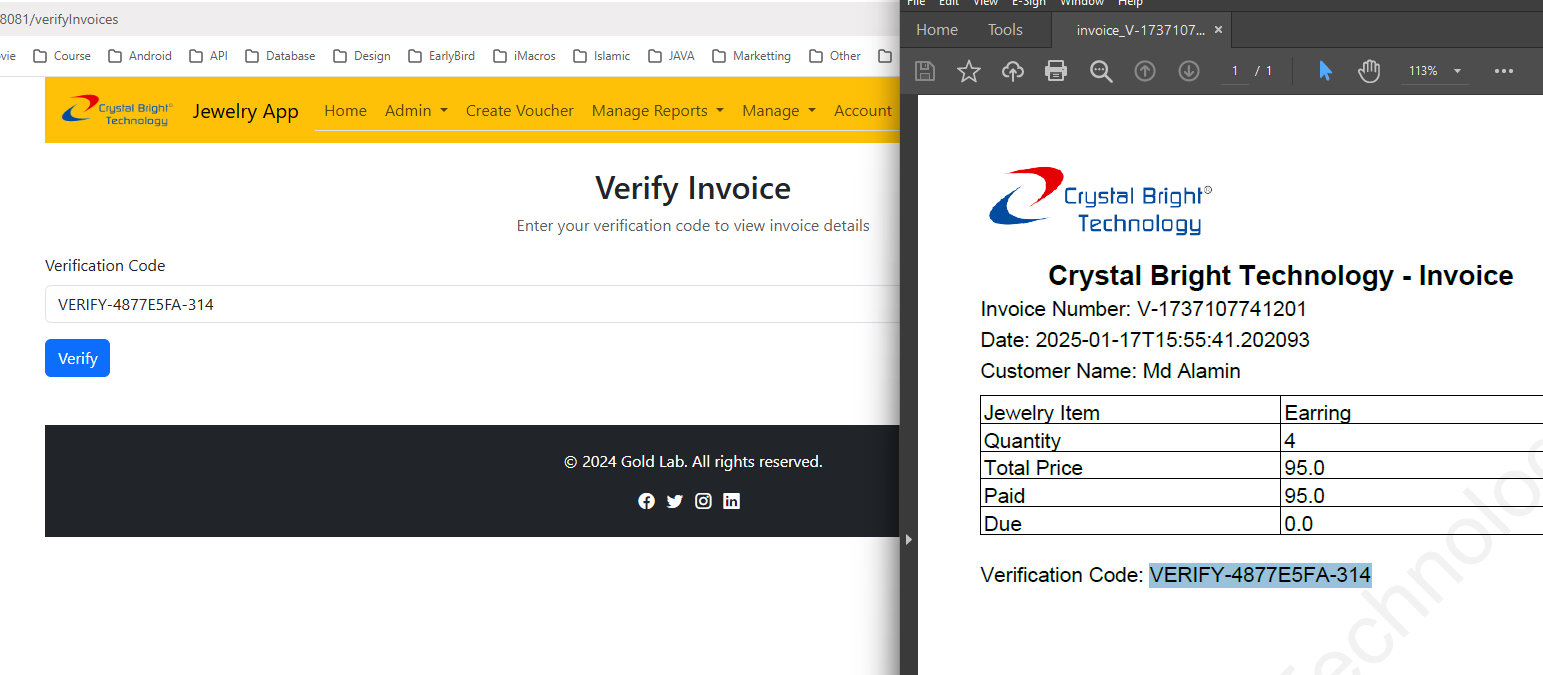


Figure 17 - Verify Invoice Over online

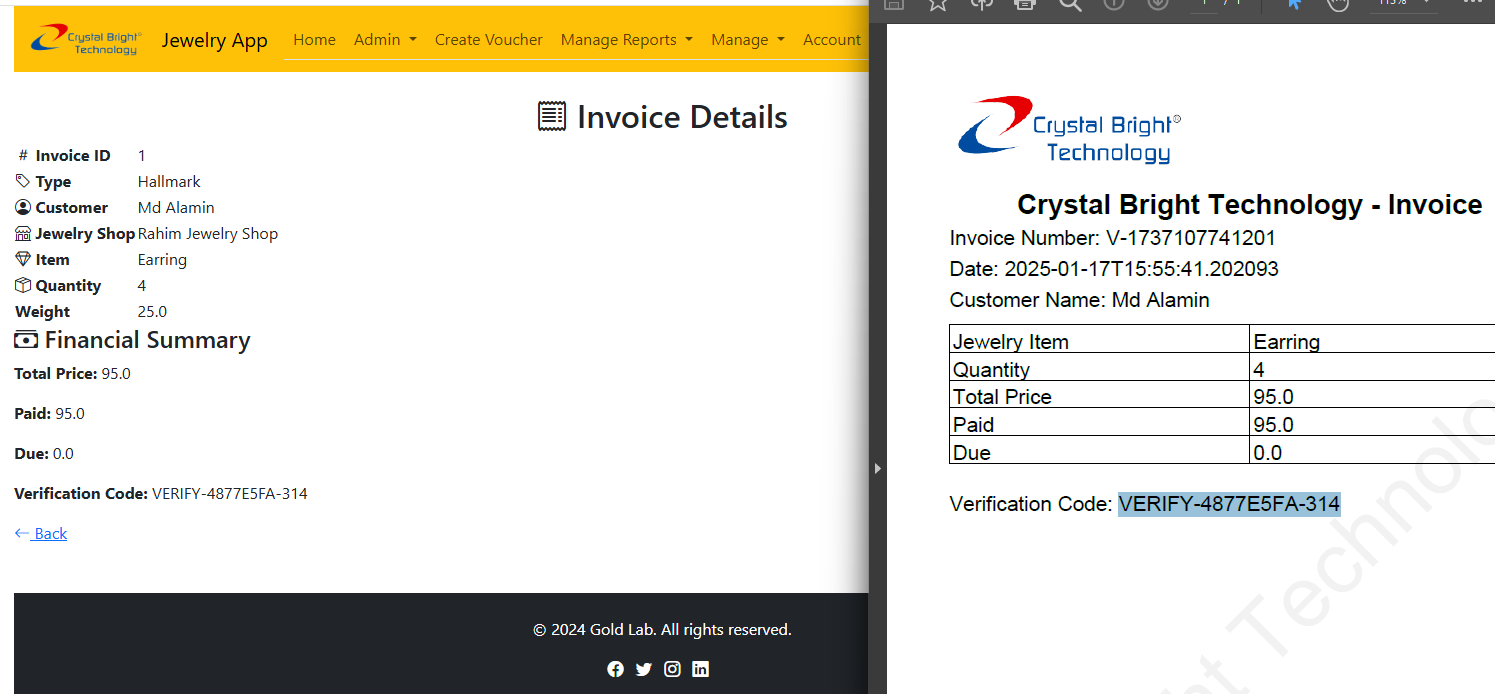


Figure 18 - Matching Invoice over online

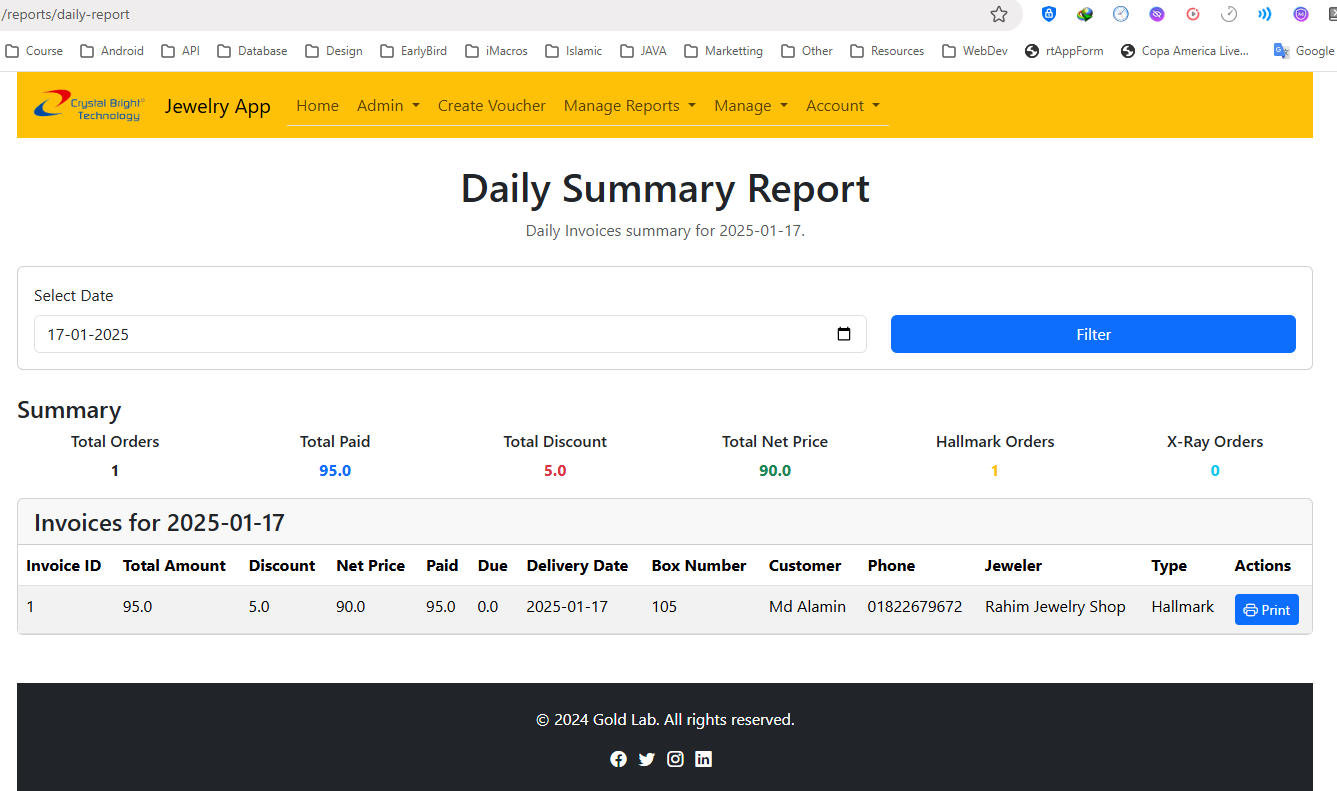


Figure 19 - Daily Summary Report

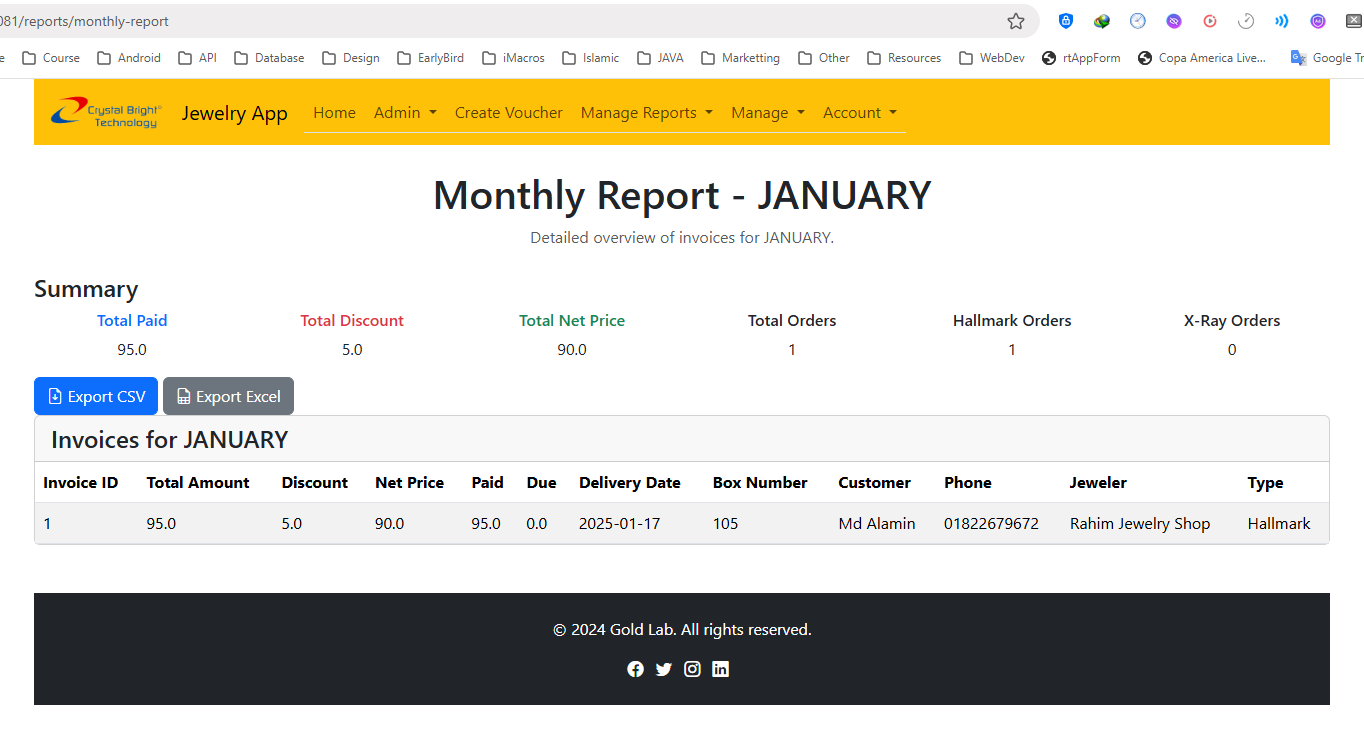


Figure 20 - Monthly Summary Report

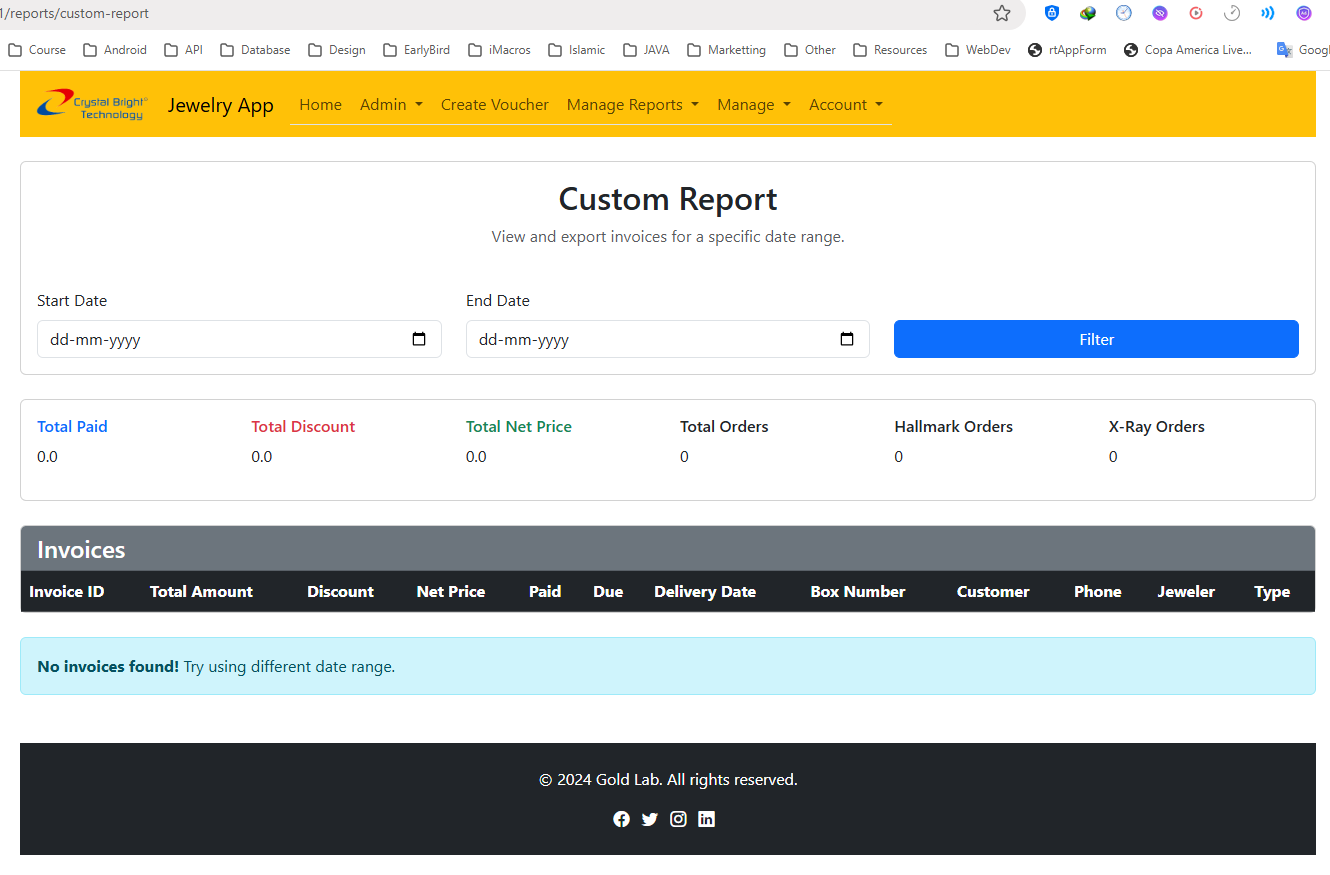


Figure 21 - Custom Filtering Based Summary Report

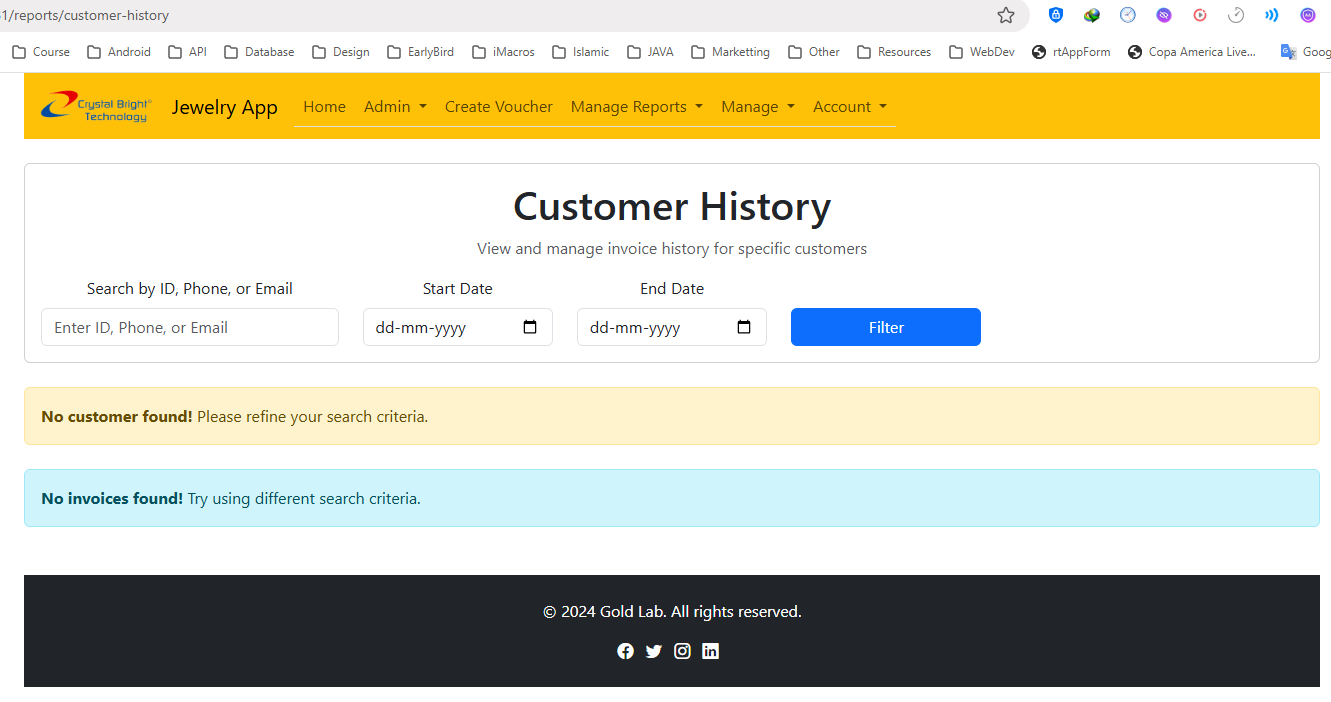


Figure 22 - Find Customer's (invoice) History

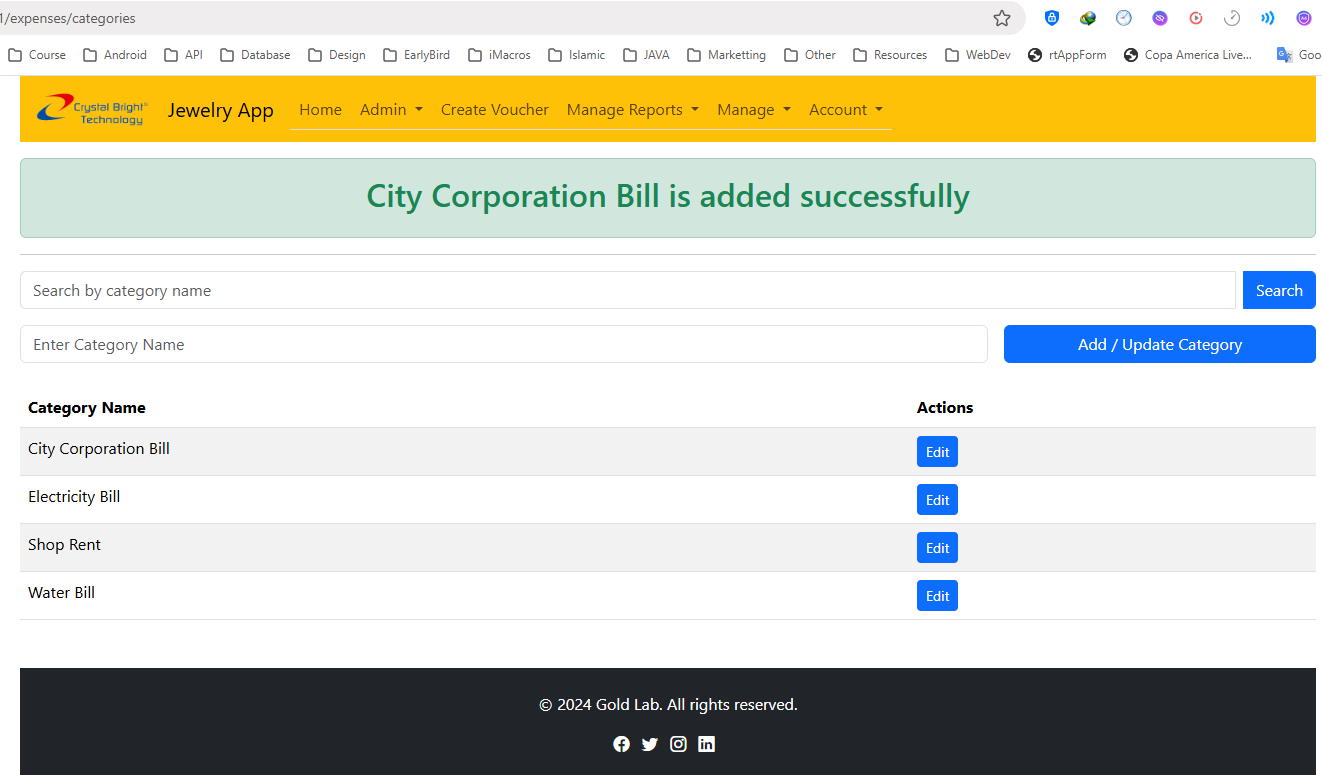


Figure 23 - Expense Category CRUD

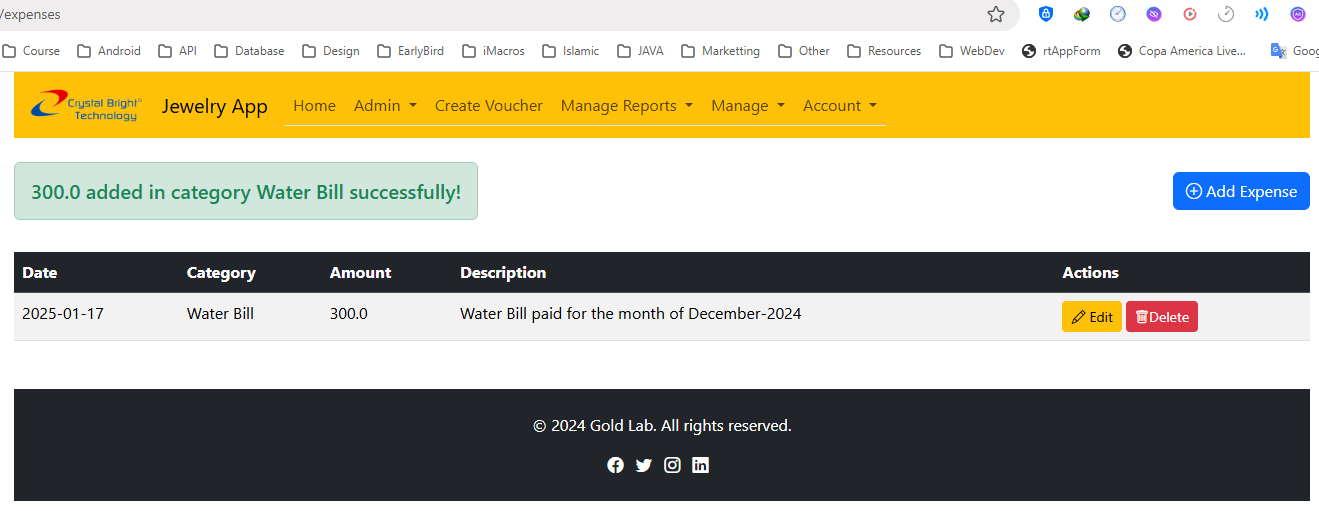


Figure 24 - Expense List

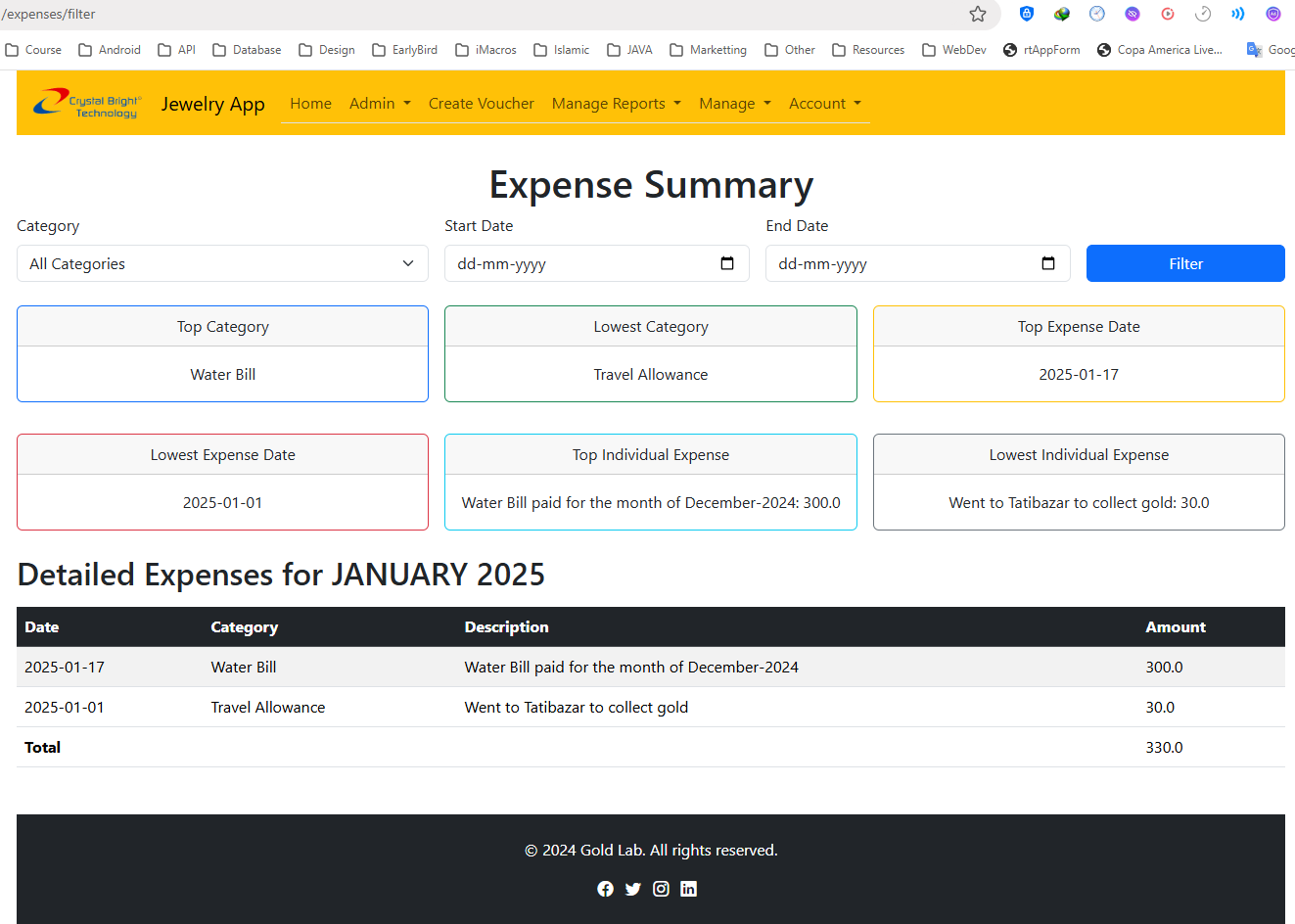


Figure 25 - Expense Summary Filtered Based

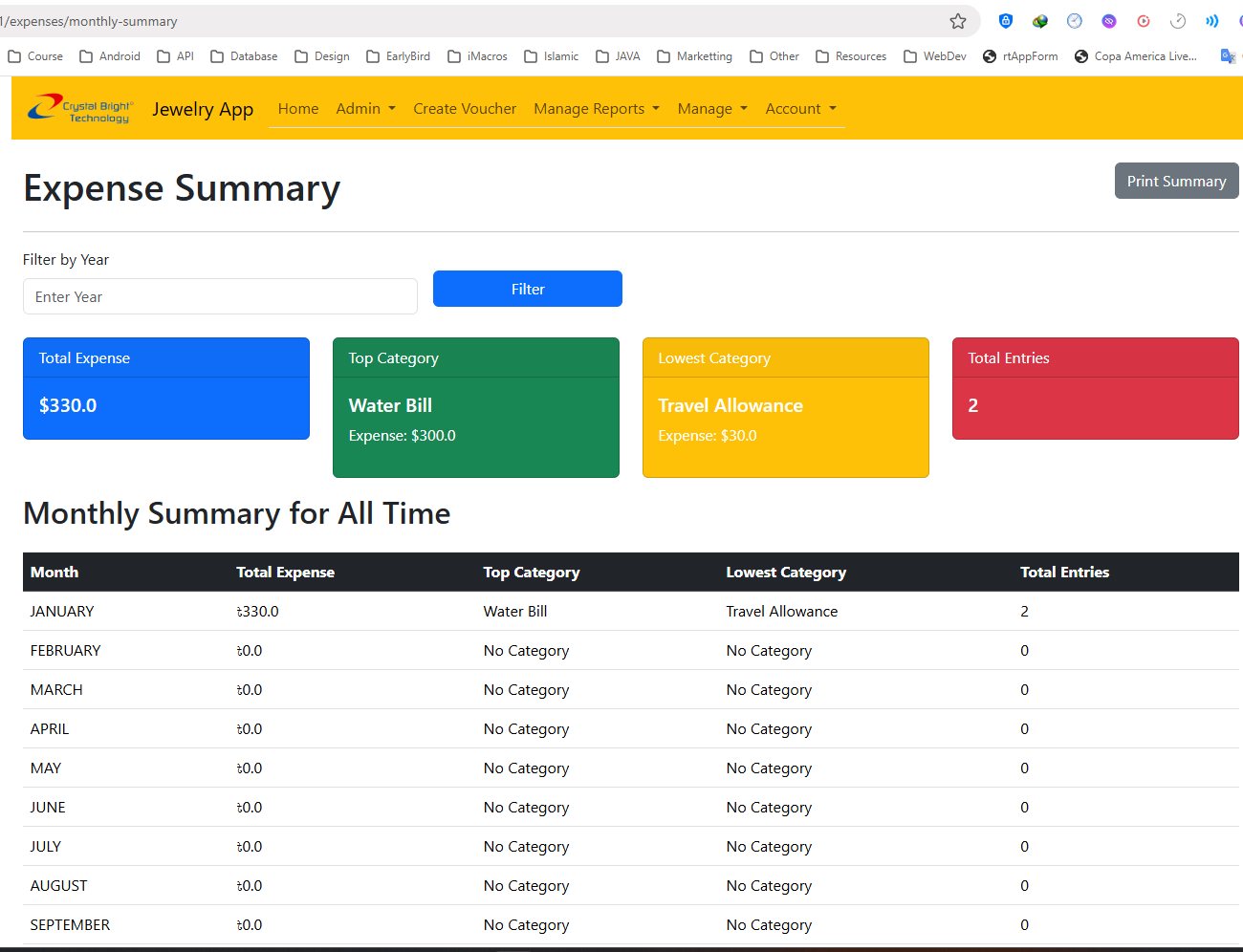


Figure 26 - Expense Monthly Summary by Year wise

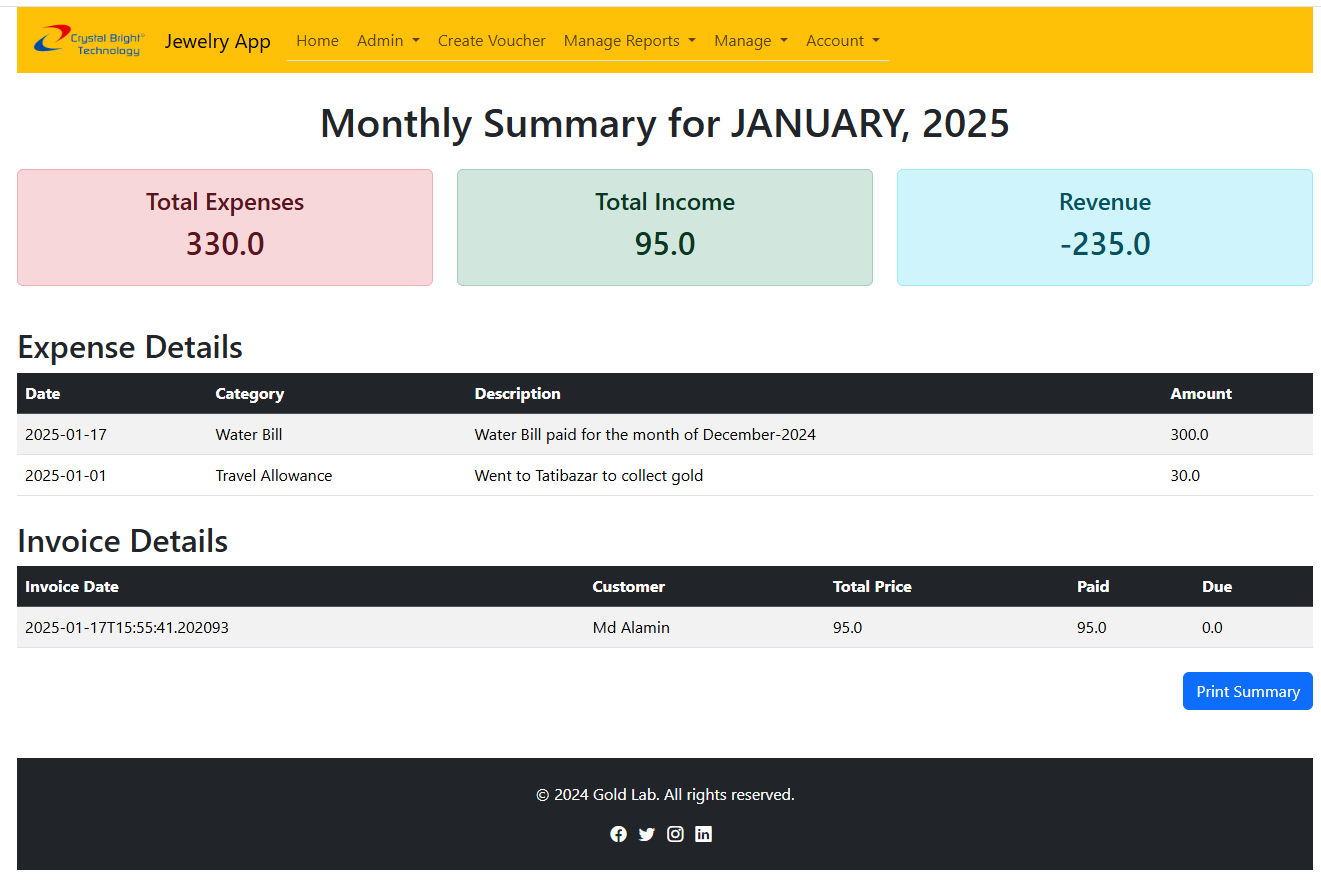


Figure 27 – Current Month Revenue Summary

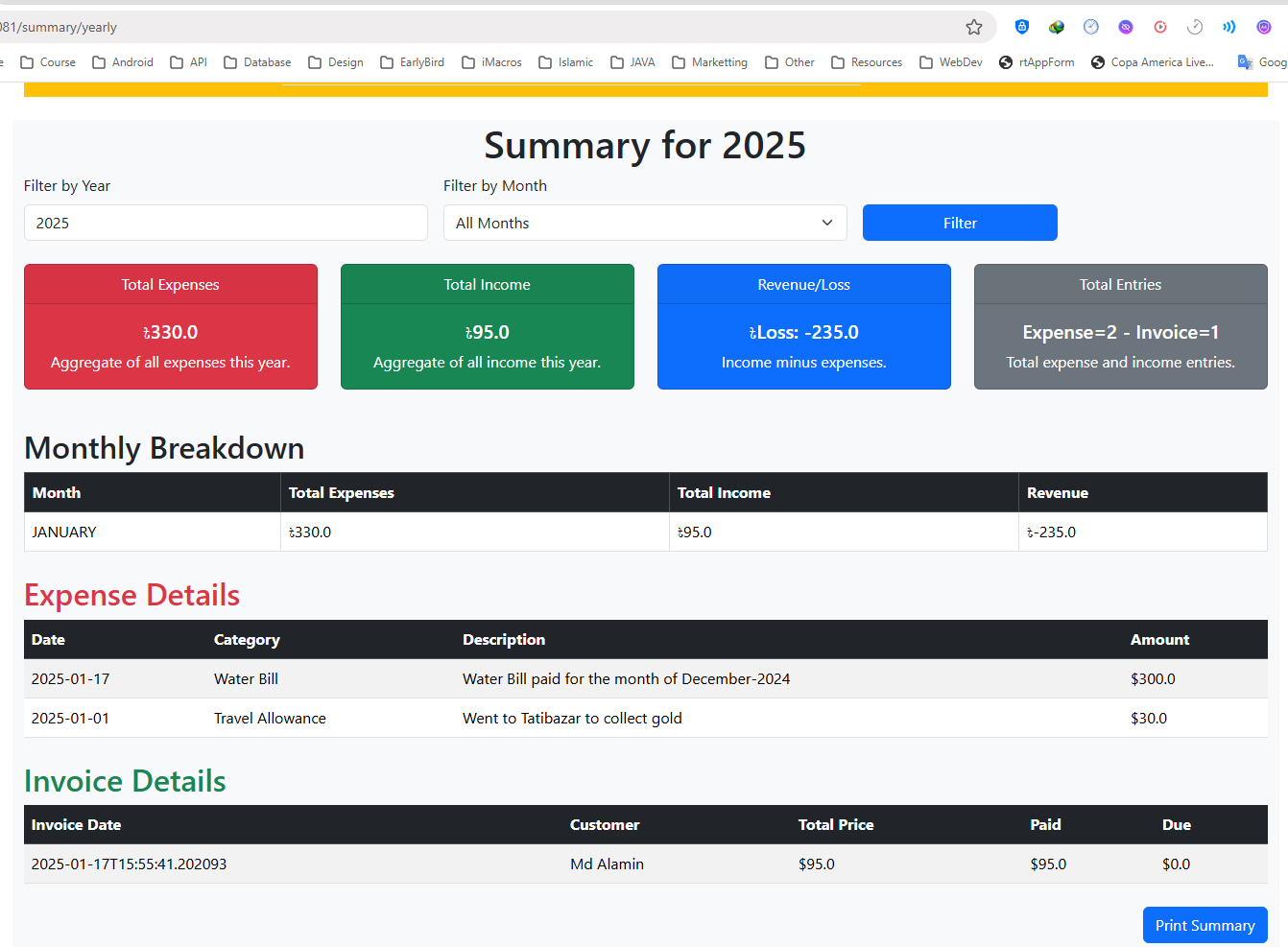


Figure 28 - Filter Based Revenue Summary

# 7. Conclusion & Upcoming Features:

## 7.1 Conclusion

The **Gold Lab Management System (GoldLabCBT)** is a robust solution designed to automate and streamline operations in jewelry businesses. By integrating customer management, invoicing, expense tracking, and revenue summaries, the system delivers a centralized and efficient platform for managing daily operations.

Key achievements of this system include:

1.**Enhanced Productivity**:   
 o Eliminates manual processes, saving time and reducing effort. o Allows staff to focus on customer service and other critical tasks.

2.**Improved Accuracy**:   
 o Minimizes errors in calculations, invoicing, and report generation.

o Ensures consistent data storage and retrieval.

3.**Real-Time Insights**:   
 o Provides comprehensive dashboards and reports.

o Supports informed decision-making through advanced revenue and expense analysis.

4.**User-Friendly Interface**:   
 o Developed with Bootstrap 5.3.3, the system offers a clean, modern, and responsive UI.

o Easy navigation ensures accessibility for all users.

This project demonstrates the integration of modern tools and frameworks like Spring Boot, MySQL, and Thymeleaf to create an application that is both scalable and reliable.

## 7.2. Upcoming Features

To stay ahead in a competitive market, the system is designed to evolve with future enhancements. Upcoming features include:

1.**Mobile App Integration**:   
 o Develop a mobile application for Android and iOS platforms.

o Enable users to manage invoices, expenses, and customer data on the go.

2.**Advanced Analytics Using AI**:   
 o Implement AI algorithms for predictive analytics.

o Provide insights into sales trends, expense patterns, and customer behavior.

3.**Integration with Payment Gateways**:

o Enable seamless integration with popular payment gateways like PayPal, Stripe, and local

providers.

o Support direct invoice payments and expense settlements online.

These future enhancements will further solidify Gold Lab Management System's position as a cutting-edge solution in the jewelry management domain.

# References

The project relied on the following resources for development and implementation:

* **Spring Boot Documentation**:
  + <https://spring.io/>
* **Bootstrap 5.3.3 Documentation**:
  + <https://getbootstrap.com/docs/5.3/getting-started/download/>
* **MySQL Documentation**:
  + <https://dev.mysql.com/doc/>
* **Maven Tools and Plugin**
  + <https://mvnrepository.com/>

# Bibliography

1.**"Java Persistence with Hibernate"**:

o A foundational resource for designing and managing relational databases with Hibernate.

2.**"Spring Boot in Action 6ed by Craig Walls"**:

o Detailed insights into building applications with Spring Boot.

3.**Online Blogs and Tutorials**:

o Practical knowledge and solutions for real-world challenges during development such as making PDF Invoice, Excel/CSV format.

# Appendices

**Task Contribution Based on SDLC Phase Table:**

|  |  |  |  |
| --- | --- | --- | --- |
| **SDLC Phase** | **Task** | **Contributor(s)** | **Description of Contribution** |
| **Requirement Gathering, and Planning** | Requirement Analysis | Afifa, Nishpa | Collaborated to gather system  requirements and analyze the manual processes in existing systems. |
| Feasibility Study | Md. Alamin | Conducted a feasibility study covering technical, economic, and operational aspects. |
| **System Design** | Logical Design | Afifa, Nishpa | Designed entity relationships and logical flow for system processes. |
| Physical Design | Md. Alamin | Implemented the database structure, linking all necessary entities. |
| **Development** | Backend  Development | Md. Alamin, | Developed Spring Boot services and  controllers for handling business logic and requests. |
| Frontend  Development | Afifa, Nishpa | Designed and developed responsive pages using Thymeleaf and Bootstrap. |
| **Testing** | Unit Testing | Md. Alamin | Performed unit testing for backend services and ensured functionality. |
| Integration Testing | Nishpa | Validated the integration between modules (invoices, expenses, and revenue  summaries). |
| **Implementation and Training** | Deployment and Training | All Members | Set up MySQL database and deployed the system on a local server for testing. |
| **Maintenance and Future Planning** | Documentation | All Members | Contributed to creating project  documentation, including the report and user manual. |
| Future Feature Planning | Nishpa | Suggested AI-based analytics and mobile app integration for upcoming releases. |