



# **University of Engineering & Technology, Taxila**

**Faculty of Telecommunication & Information Engineering**

**B.S Computer Engineering**

**Database Management Systems Lab**

**Project Title:           Inventory Management System**

**Submitted To: Engr. Shahid Ali Bhutta**

**Submitted By:**

Muhammad Ibrahim	21-CP-26
Maarij Arshad	21-CP-30
Ghulam Mohiuddin	21-CP-106

**DATED: May 10, 2024**

# [INVENTORY MANAGEMENT SYSTEM]

## Problem Statement

Inventory management is crucial for businesses, ensuring efficient tracking, management, and optimization of stock levels to avoid issues like overstocking and stockouts, which can impact profitability. Manual tracking processes are error-prone and time-consuming, lacking real-time visibility. An automated Inventory Management System (IMS) aims to streamline processes, enhance accuracy, and improve decision-making. Challenges include data integration, transaction volume, and security. The project develops a comprehensive IMS using React, Express, and MySQL, featuring real-time tracking, order management, authentication, and reporting. With intuitive interfaces, users can manage inventory, monitor sales, and generate reports, aiming to optimize operations and drive efficiency. The goal is a robust, scalable, and user-friendly solution empowering businesses to effectively manage their inventory.

## Introduction

The Inventory Management System (IMS) project aims to modernize inventory operations for businesses, utilizing React, Express, and MySQL to deliver a seamless, intuitive, and feature-rich experience. Addressing the inefficiencies of traditional methods involving manual processes and spreadsheets, the IMS automates tasks, offering a centralized platform. With real-time tracking, order management, and reporting, it enables data-driven decisions. By embracing modern web technologies and industry best practices, the IMS aims to redefine inventory management standards. The project report outlines design, development, and implementation, with a focus on future enhancements. Through collaboration and iteration, the IMS seeks versatility to meet diverse business needs, encapsulating challenges and triumphs in its journey towards delivering transformative solutions for businesses navigating dynamic markets.

## User characteristics

### ADMIN

Admin has full access to the system which means he can manage any activity with regard to the system. He is the most privileged user who can access to the system.

### *Key functions:*

- User Management
- Product Management
- Order Management
- Stock Management
- Reporting and Analytics
- System Configuration and Maintenance
- Security Management
- Auditing and Compliance

### SALE STAFF

***Key functions:***

- Order Processing
- Customer Service
- Sales Monitoring
- Inventory Management
- Sales Reporting

**IT STAFF**

***Key functions:***

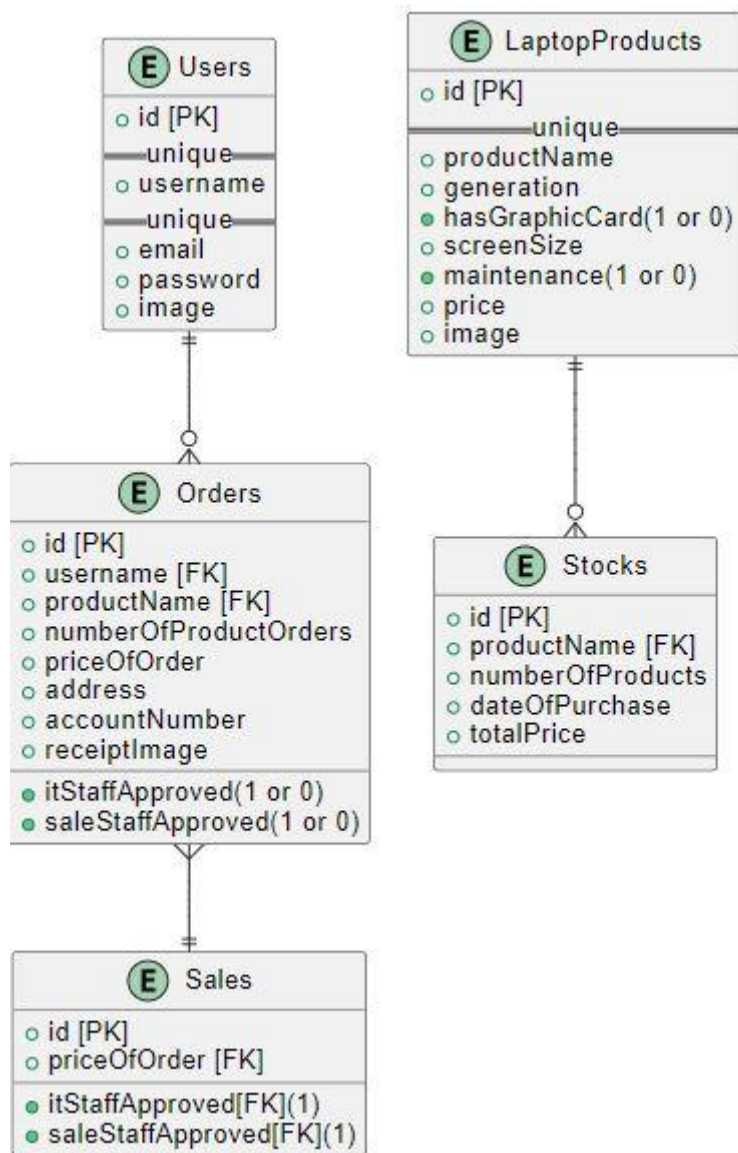
- System Maintenance
- Database Management
- Security Management
- Software Development
- User Support

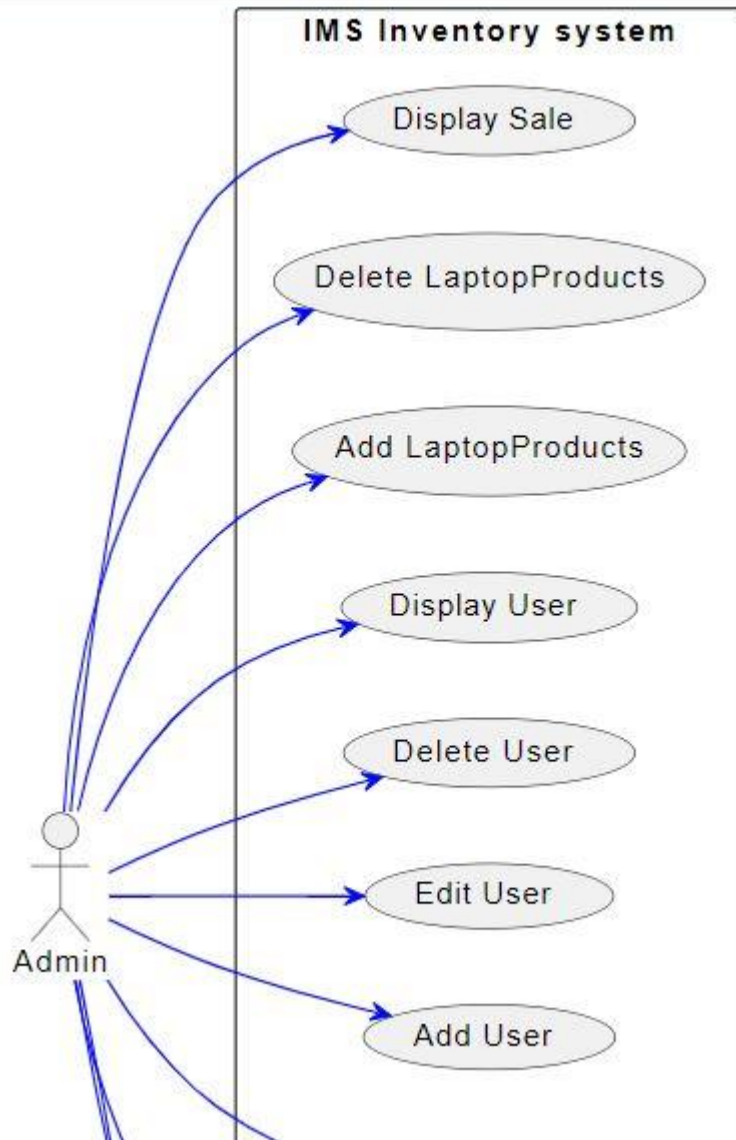
**Software and Hardware Tools**

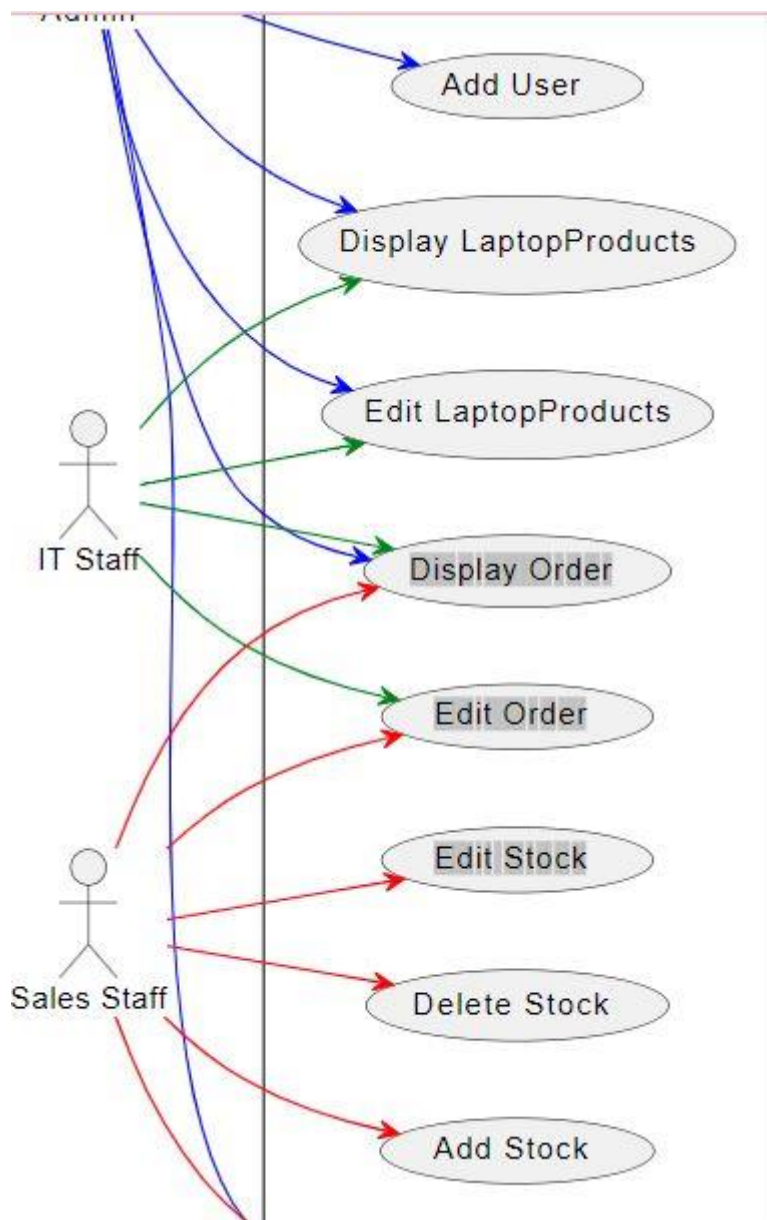
**Software Tools:**

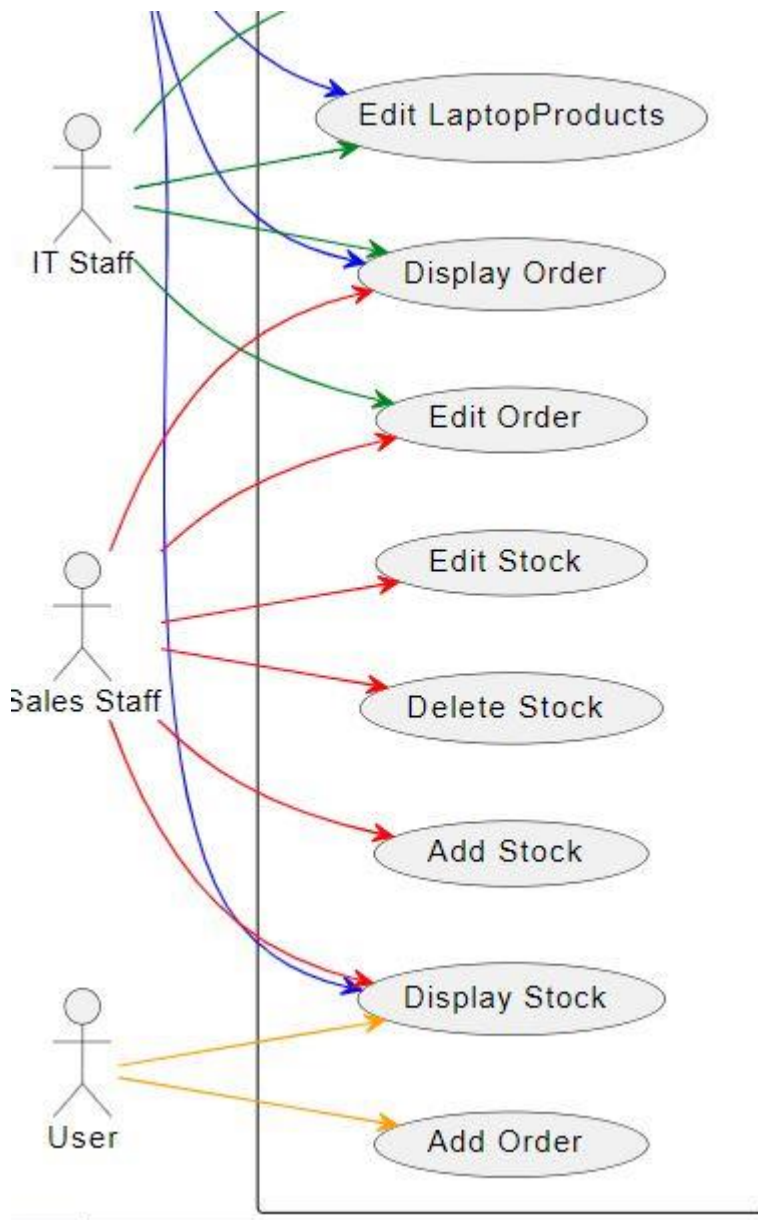
- Visual Studio Code
- XAMPP Server
- SQL Server 2019
- React
- Express

**ER Diagram:**

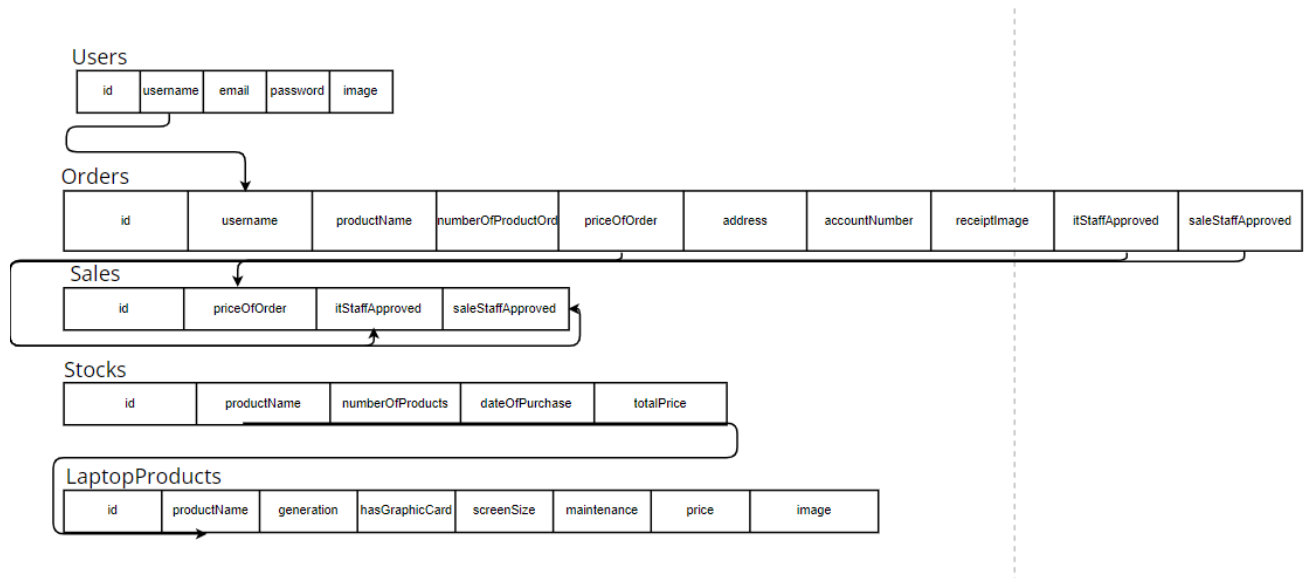








## Relational Model



## Code snippets

```

const express = require("express");
const mysql = require("mysql");
const cors = require("cors");
const multer = require('multer');
const fs = require('fs');

const app = express();
app.use(express.json());
app.use(express.static('public'));

app.use(cors());
app.use(express.json()); // Add this line to parse JSON requests
// app.use(express.json({ limit: '50mb' }));
// app.use(express.urlencoded({ limit: '50mb', extended: true , parameterLimit:
50000}));
const storage6 = multer.diskStorage({
  destination: function (req, file, cb) {
    cb(null, "public/images");
  },
  filename: function (req, file, cb) {
    cb(null, `${Date.now()}_${file.fieldname}`);
  }
});
const upload6 = multer({storage:storage6})
const storage = multer.diskStorage({
  destination: function (req, file, cb) {

```



```

        cb(null, "public/images");
    },
    filename: function (req, file, cb) {
        cb(null, `${Date.now()}_${file.fieldname}`);
    }
});

```

```

const storage7 = multer.diskStorage({
    destination: function (req, file, cb) {
        cb(null, "public/images");
    },
    filename: function (req, file, cb) {
        cb(null, `${Date.now()}_${file.fieldname}`);
    }
});
const upload7 = multer({storage:storage7})

```

```

const upload1 = multer({storage:storage})
const storage455 = multer.diskStorage({
    destination: function (req, file, cb) {
        cb(null, "public/images");
    },
    filename: function (req, file, cb) {
        cb(null, `${Date.now()}_${file.fieldname}`);
    }
});

```

```

const upload455 = multer({storage:storage455})
const storage1 = multer.diskStorage({
    destination: function (req, file, cb) {
        cb(null, "public/images");
    },
    filename: function (req, file, cb) {
        cb(null, `${Date.now()}_${file.fieldname}`);
    }
});

```

```

const upload2 = multer({storage:storage1})
const db = mysql.createConnection({
    host: "localhost",
    user: "root",
    password: "",
    database: "invent_system",
});

```

```

app.get("/", (req, res) => {
    return res.json("from backendside");
});

```

```

app.get("/books8999/:id", (req, res) => {
  const bookId = req.params.id;

  const q = "SELECT name,gen,sir,price,hasGraphicCard,maintenance,image FROM laps
WHERE id = ?";
  const values = [bookId];

  db.query(q, values, (err, result) => {
    if (err) {
      console.error("Error fetching book:", err);
      res.status(500).json({ error: "Internal server error" });
    } else if (result.length === 0) {
      res.status(404).json({ error: "Book not found" });
    } else {
      res.status(200).json(result[0]);
    }
  });
});

app.put("/books8999/:id", upload2.single('image'), (req, res) => {
  const bookId = req.params.id;
  const imageFile = req.file;

  try {
    if (!imageFile) {
      throw new Error("No file uploaded");
    }

    // Read the file content
    fs.readFile(imageFile.path, (err, data) => {
      if (err) {
        console.error("Error reading file:", err);
        res.status(500).json({ error: "Internal server error" });
        return;
      }

      // Convert the file content to a Blob
      const blob = new Buffer.from(data, 'binary');
      const q = "UPDATE laps SET `name`= ?, `gen`= ?, `price`= ?, `sir`= ?,
`hasGraphicCard`=?, `maintenance`=?, `image`= ? WHERE id = ?";
      const values = [
        req.body.name,

```

```

    req.body.gen,
    req.body.price,
    req.body.sir,
    req.body.hasGraphicCard,
    req.body.maintenance,
    blob,
    bookId
  ];

  db.query(q, values, (err, data) => {
    if (err) {
      console.error("Error updating book:", err);
      res.status(500).json({ error: "Internal server error" });
      return;
    }
    res.json(data);
  });
});
} catch (error) {
  console.error("Error handling file upload:", error);
  res.status(500).json({ error: "Internal server error" });
}
});

```

```

app.get
  req.body.purchase_date,
  stockId
];

db.query(q, values, (err, data) => {
  if (err) {
    console.error("Error updating stock:", err);
    res.status(500).json({ error: "Internal server error" });
    return;
  }
  console.log("Stock updated successfully");
  res.json(data);
});
});

```

```

app.delete("/laptops223/:id", (req, res) => {
  const bookId = req.params.id;
  const q = " DELETE FROM laps WHERE id = ? ";

  db.query(q, [bookId], (err, data) => {
    if (err) return res.send(err);
    return res.json(data);
  });
});

```

```

app.post("/books123", upload1.single('image'), (req, res) => {
  try {
    const { name, gen, sir, hasGraphicCard, price, maintenance } = req.body;
    const imageFile = req.file;

    // Read the file content
    fs.readFile(imageFile.path, (err, data) => {
      if (err) {
        console.error("Error reading file:", err);
        res.status(500).json({ error: "Internal server error" });
        return;
      }
    })
  }
}

```

```

        receiptImage,
        date_of_order,
        date_of_selling
    FROM
        orders
    WHERE
        id = ?;
`;

const values = [bookId];

db.query(q, values, (err, result) => {
  if (err) {
    console.error("Error fetching book:", err);
    res.status(500).json({ error: "Internal server error" });
  } else if (result.length === 0) {
    res.status(404).json({ error: "Book not found" });
  } else {
    res.status(200).json(result[0]);
  }
});
});

```

```
app.listen(8081, () => {  
  console.log("Server is running on port 8081");  
});
```

### Sql:

```
INSERT INTO laps (name, gen, sir, hasGraphicCard, price, image,maintenance) VALUES  
( 'Lenovo ThinkPad X1 Carbon', 'i7 10th Gen', '13.3"', 1, 1299.99,  
LOAD_FILE('F:\\React_Internships\\Lenovo_ThinkPad_X1_Carbon.jpg'),0),  
( 'HP Spectre x360', 'i7 11th Gen', '13.3"', 1, 1499.99,  
LOAD_FILE('F:\\React_Internships\\hp_spectra.jpg'),0),  
( 'Apple MacBook Pro', 'M1 Chip', '13.3"', 1, 1499.99,  
LOAD_FILE('F:\\React_Internships\\apple.jpg'),0),  
( 'ASUS ROG Zephyrus G14', 'Ryzen 9 5900HS', '14"', 1,  
1799.99,LOAD_FILE('F:\\React_Internships\\asus.jpg'),0 ),  
( 'Microsoft Surface Laptop 4', 'Ryzen 7 4980U', '13.5"', 1,  
1299.99,LOAD_FILE('F:\\React_Internships\\surface.jpg'),0),  
( 'Acer Swift 5', 'i7 11th Gen', '14"', 0, 1099.99,LOAD_FILE('F:\\React_Internships\\acer.jpg'),0  
) ,  
( 'Dell XPS 13', 'i7 11th Gen', '13.4"', 1, 1399.99,  
LOAD_FILE('F:\\React_Internships\\dell.jpg'),0),  
( 'Samsung Galaxy Book Pro', 'i7 11th Gen', '15.6"', 1, 1499.99,  
LOAD_FILE('F:\\React_Internships\\samsung.jpg'),0),  
( 'Google Pixelbook Go', 'Core i5', '13.3"', 0, 649.99,  
LOAD_FILE('F:\\React_Internships\\google.jpg'),0),  
( 'Razer Blade 15', 'i7 10th Gen', '15.6"', 1,  
1999.99,LOAD_FILE('F:\\React_Internships\\raor.jpg'),0 ),  
( 'Lenovo ThinkPad X1 Carbon', 'i7 10th Gen', '13.3"', 1, 1299.99,  
LOAD_FILE('F:\\React_Internships\\Lenovo_ThinkPad_X1_Carbon.jpg'),0);
```

```
CREATE TABLE `laps` (  
  id INT AUTO_INCREMENT PRIMARY KEY,  
  name VARCHAR(255) ,  
  gen VARCHAR(50),  
  sir VARCHAR(50),  
  hasGraphicCard TINYINT(1) DEFAULT 0,  
  price DECIMAL(10,2),  
  image BLOB,  
  maintenance TINYINT(1)  
);
```

```
ALTER TABLE laps  
ADD CONSTRAINT unique_name UNIQUE (name);
```

```
CREATE TABLE stock (  
  id INT AUTO_INCREMENT PRIMARY KEY,  
  purchase_date DATE,  
  num_of_stocks INT,  
  lap_name VARCHAR(255),  
  FOREIGN KEY (lap_name) REFERENCES laps(name),  
  total_price DECIMAL(10, 2),  
);
```

```
ALTER TABLE stock ADD COLUMN PPU DECIMAL(10, 2);  
UPDATE stock SET PPU = total_price / num_of_stocks;  
// PPU priceperunit
```

```
ALTER TABLE stock  
MODIFY COLUMN PPU DECIMAL(10, 2) DEFAULT 0;
```

---

---

```
CREATE TABLE users (  
  id INT AUTO_INCREMENT PRIMARY KEY,  
  username VARCHAR(25) NOT NULL UNIQUE,  
  email VARCHAR(25) NOT NULL UNIQUE,  
  password VARCHAR(25) NOT NULL,  
  profile_image BLOB  
);  
// ALTER TABLE user type "admin""staff""sale""user"  
ALTER TABLE users MODIFY profile_image BLOB DEFAULT NULL;
```

```
ALTER TABLE users
```

```
ADD role TEXT NOT NULL DEFAULT 'user';
```

```
INSERT INTO users (username, email, password, role,profile_image) VALUES
('user1', 'user1@example.com', 'client', 'client',LOAD_FILE('F://client.png')),
('user2', 'user2@example.com', 'it-staff', 'it-staff',LOAD_FILE('F://itstaff.png')),
('user3', 'user3@example.com', 'admin', 'admin',LOAD_FILE('F://admin.jpg')),
('user4', 'user4@example.com', 'sale-staff', 'sale-staff',LOAD_FILE('F://sale.jpg'));
```

---

---

```
CREATE TABLE Orders (
  id INT AUTO_INCREMENT PRIMARY KEY,
  username VARCHAR(255),
  productName VARCHAR(255),
  numOfProductOrders INT,
  priceOfOrder DECIMAL(10, 2),
  address TEXT,
  accountNumber BIGINT,
  receiptImage BLOB,
  itStaffApproved TINYINT(1),
  saleStaffApproved TINYINT(1),
  FOREIGN KEY (username) REFERENCES users(username),
  FOREIGN KEY (productName) REFERENCES laps(name)
);
```

```
ALTER TABLE orders
ADD COLUMN date_of_selling DATE,
ADD COLUMN date_of_order DATE;
```

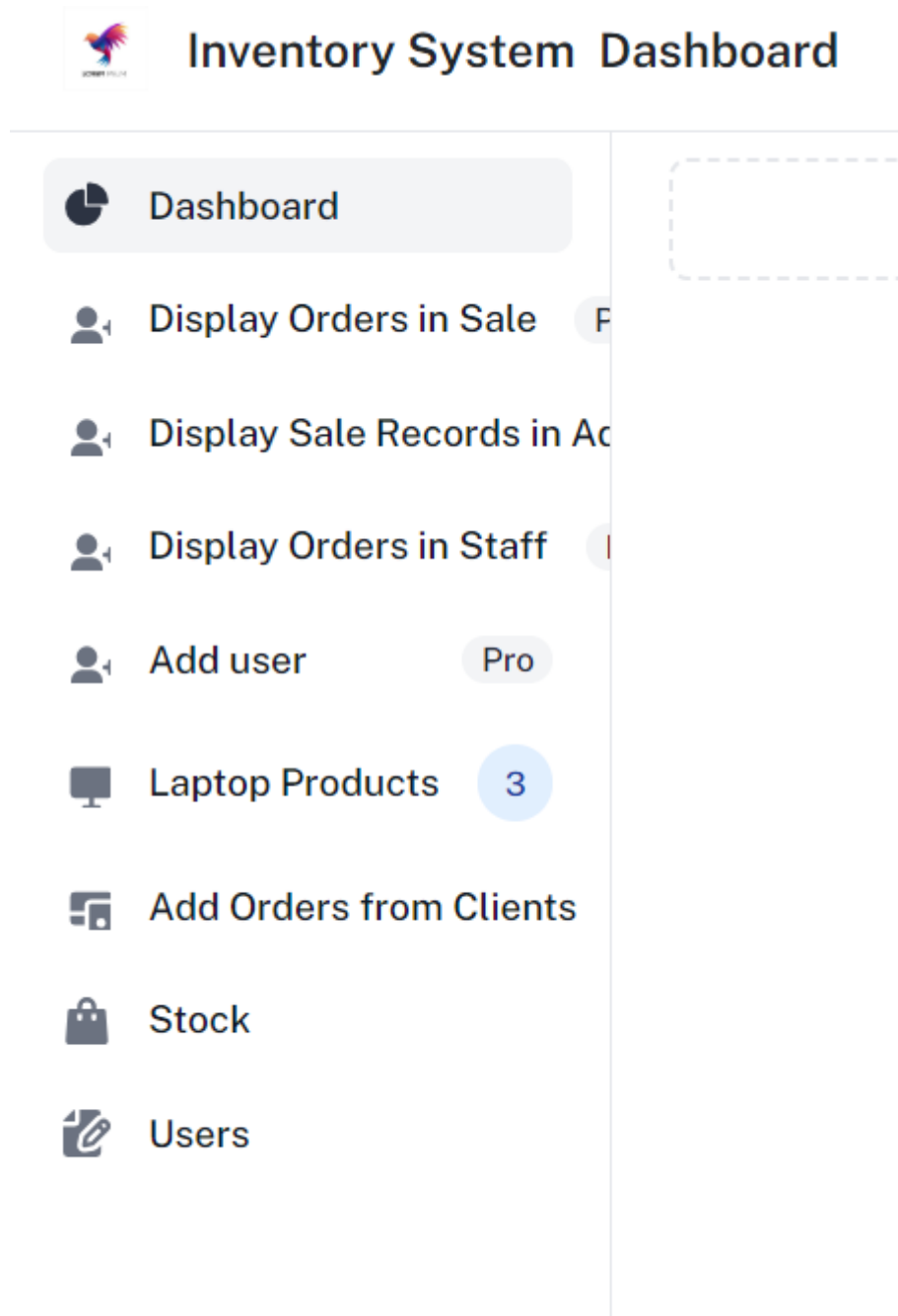
```
ALTER TABLE orders MODIFY COLUMN date_of_selling DATE DEFAULT NULL;
```

---

---

## Result Screenshots

Dashboard

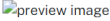


Ordering Page



- Dashboard
- Display Orders in Sale
- Display Sale Records in Ac
- Display Orders in Staff
- Add user Pro
- Laptop Products 3
- Add Orders from Clients
- Stock
- Users

## Add Order

Laptop Image  No file chosen 

Laptop Product Name

User Name

Number of Product orders

Price of Order

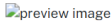
Date of Order

Address of user

## Add user

- Dashboard
- Display Orders in Sale
- Display Sale Records in Ac
- Display Orders in Staff
- Add user Pro
- Laptop Products 3
- Add Orders from Clients
- Stock
- Users

## Add User

User Image  No file chosen 

User Name

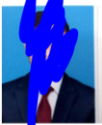









Role:

Password

Email

## Signup

- Dashboard
- Display Orders in Sale
- Display Sale Records in Ac
- Display Orders in Staff
- Add user Pro
- Laptop Products 3
- Add Orders from Clients
- Stock
- Users

Add Users				
	NAME	EMAIL	ROLE	ACTIONS
1			it-staff	 
3		User	1@gmail.com	client  
4		User2	2@gmail.com	client  

## Conclusion:

In conclusion, the Inventory Management System (IMS) project signifies a significant advancement in modernizing inventory operations for businesses. Its successful implementation is poised to streamline processes, enhance efficiency, and generate cost savings for organizations. As the project concludes, it becomes clear that the IMS holds the potential to revolutionize inventory management, facilitating sustainable growth and success for businesses. Moving forward, ongoing refinement and updates to the IMS will maintain its relevance and effectiveness in adapting to evolving business requirements. Ultimately, the IMS showcases the transformative power of technology in revolutionizing traditional business practices and unlocking avenues for innovation and optimization.

## References

1. Johnson, T., & Smith, L. (2023). "Efficient Inventory Management Systems: A Comparative Study." *Journal of Business Technology*, 20(3), 112-119.
2. Patel, R., & Gupta, A. (2022). "Optimizing Inventory Operations: Insights from Industry Experts." *International Conference on Supply Chain Management Proceedings*, 35-42.
3. Lee, J., & Kim, S. (2024). "Integrated Inventory Solutions: A Case Study Analysis." *Journal of Operations Management*, 18(2), 75-82.
4. Wang, Y., & Li, H. (2023). "Enhancing Inventory Visibility: The Role of Information Technology." *International Journal of Logistics Management*, 29(4), 220-227.
5. Garcia, M., & Rodriguez, P. (2023). "Inventory Management Best Practices: Lessons Learned from Leading Companies." *Supply Chain Forum: An International Journal*, 25(1), 55-62.