

Installment Management System (IMS) - User Guide

Table of Contents

1. [System Overview](#)
 2. [Installation Setup](#)
 3. [Database Setup](#)
 4. [Running the Application](#)
 5. [User Roles and Permissions](#)
 6. [Login Process](#)
 7. [Dashboard Features](#)
 8. [Filtering and Search](#)
 9. [Verification Management](#)
 10. [Bulk Operations](#)
 11. [Troubleshooting](#)
-

System Overview

The Installment Management System (IMS) is a web-based application designed to manage customer installment placement data with role-based access control. The system allows authorized users to view, filter, and update verification status of installment applications.

Key Features:

- **Role-based Access Control** (Reader, Editor, Admin)
 - **Advanced Filtering** (by status, merchant, date, etc.)
 - **Search Functionality** across multiple fields
 - **Individual and Bulk Verification** updates
 - **Responsive Design** for desktop and mobile use
-

Installation Setup

Prerequisites

- Node.js installed on your system
- MySQL database server running

- Git (optional, for cloning)

Step 1: Create Project Directory

```
bash  
mkdir ims-system  
cd ims-system
```

Step 2: Initialize Node.js Project

```
bash  
npm init -y
```

Step 3: Install Dependencies

```
bash  
npm install express mysql2 express-session
```

Step 4: Create Directory Structure

```
bash  
mkdir public
```

Step 5: Create Files

1. Create `app.js` in the root directory (copy the server code)
2. Create `public/login.html` (copy the login page code)
3. Create `public/dashboard.html` (copy the dashboard code)

Database Setup

Step 1: Update Database Configuration

Ensure your database configuration in `app.js` matches your MySQL setup:

```
javascript
```

```
const dbConfig = {
  host: 'localhost',
  database: 'IMS_throwhorse',
  user: 'root',
  password: 'Root_123',
  port: 3307,
  waitForConnections: true,
  connectionLimit: 10,
  queueLimit: 0
};
```

Step 2: Create Database Tables

Create the installment_applications table:

```
sql

CREATE TABLE installment_applications (
  id INT AUTO_INCREMENT PRIMARY KEY,
  card_number VARCHAR(16) NOT NULL,
  nic VARCHAR(20) NOT NULL,
  full_name VARCHAR(100) NOT NULL,
  contact_number VARCHAR(15) NOT NULL,
  email VARCHAR(100) NOT NULL,
  merchant VARCHAR(100) NOT NULL,
  installment_plan INT NOT NULL,
  bill_amount DECIMAL(10,2) NOT NULL,
  is_verified BOOLEAN DEFAULT FALSE,
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

Create the login_credentials table:

```
sql

CREATE TABLE login_credentials (
  user_id VARCHAR(50) PRIMARY KEY,
  password VARCHAR(255) NOT NULL,
  role_access ENUM('reader', 'editor', 'admin') NOT NULL
);
```

Step 3: Add Sample User Accounts

```
sql

-- Admin user
INSERT INTO login_credentials (user_id, password, role_access)
VALUES ('admin', 'admin123', 'admin');

-- Editor user
INSERT INTO login_credentials (user_id, password, role_access)
VALUES ('editor', 'edit123', 'editor');

-- Reader user
INSERT INTO login_credentials (user_id, password, role_access)
VALUES ('reader', 'read123', 'reader');
```

Step 4: Add Sample Data (Optional)

```
sql

INSERT INTO installment_applications
(card_number, nic, full_name, contact_number, email, merchant, installment_plan, bill_amount)
VALUES
('1234567890123456', '123456789V', 'John Doe', '0771234567', 'john@email.com', 'ABC Store', 12, 50000.00),
('2345678901234567', '234567890V', 'Jane Smith', '0772345678', 'jane@email.com', 'XYZ Shop', 6, 25000.00),
('3456789012345678', '345678901V', 'Bob Johnson', '0773456789', 'bob@email.com', 'DEF Mart', 24, 75000.00);
```

Running the Application

Step 1: Start the Server

```
bash

node app.js
```

Step 2: Access the Application

Open your web browser and navigate to:

```
http://localhost:3000
```

Expected Output:

Server running on <http://localhost:3000>

Please make sure to create the public directory with HTML files

User Roles and Permissions

Reader Role

- **Access Level:** View Only
- **Permissions:**
 - View all installment applications
 - Use all filtering and search features
 - Export data (if implemented)
- **Restrictions:**
 - Cannot modify verification status
 - Cannot perform bulk operations
 - No access to admin functions

Editor Role

- **Access Level:** View and Edit
- **Permissions:**
 - All Reader permissions
 - Toggle individual verification status
 - Update application verification
- **Restrictions:**
 - Cannot perform bulk operations
 - No access to admin-only functions

Admin Role

- **Access Level:** Full Access
- **Permissions:**
 - All Editor permissions
 - Bulk verify/unverify operations
 - Select multiple applications

- Advanced administrative functions
 - **Restrictions:**
 - None within the application scope
-

Login Process

Step 1: Access Login Page

Navigate to `http://localhost:3000` to access the login page.

Step 2: Enter Credentials

- **User ID:** Enter your assigned user ID
- **Password:** Enter your password

Step 3: Sign In

Click the "Sign In" button to authenticate.

Step 4: Role Verification

After successful login, you'll be redirected to the dashboard where your role will be displayed in the header.

Default Test Accounts:

User ID	Password	Role
admin	admin123	Admin
editor	edit123	Editor
reader	read123	Reader

Dashboard Features

Header Section

- **Logo:** IMS Dashboard branding
- **User Info:** Displays current user and role
- **Logout Button:** Securely end your session

Filter Panel

- **Status Filter:** Unverified/Verified/All applications

- **Merchant Filter:** Filter by specific merchant
- **Installment Plan:** Filter by plan duration
- **Date Range:** Filter by application creation date
- **Search Box:** Search across multiple fields
- **Action Buttons:** Apply filters or clear all

Data Table

- **Application Details:** All relevant customer information
- **Status Indicators:** Visual verification status
- **Action Buttons:** Individual verification controls
- **Selection Checkboxes:** For bulk operations (Admin only)

Pagination

- **Page Navigation:** Browse through large datasets
 - **Items Per Page:** Configurable display limits
 - **Total Count:** Shows filtered results summary
-

Filtering and Search

Status Filtering

- **Unverified Only** (Default): Shows applications pending verification
- **Verified Only:** Shows completed/verified applications
- **All Applications:** Shows entire dataset

Advanced Filters

Merchant Filter

- Select from dropdown of all available merchants
- Filter applications by specific merchant

Installment Plan Filter

- Select from available installment plans (6, 12, 24 months)
- Filter by payment duration

Date Range Filter

- **From Date:** Start date for filtering
- **To Date:** End date for filtering
- Filter applications by creation date

Search Functionality

Search across multiple fields simultaneously:

- Card Number
- NIC Number
- Full Name
- Contact Number
- Email Address

Using Filters Effectively

1. **Start Broad:** Begin with status filter (Unverified)
 2. **Add Specifics:** Layer additional filters as needed
 3. **Use Search:** For quick customer lookup
 4. **Apply Filters:** Click "Apply Filters" to update results
 5. **Clear When Needed:** Use "Clear All" to reset
-

Verification Management

Individual Verification (Editor/Admin)

For Unverified Applications:

1. Locate the application in the table
2. Click the green "Verify" button
3. Status will update to "Verified" immediately
4. Success notification will appear

For Verified Applications:

1. Locate the application in the table
2. Click the yellow "Unverify" button
3. Status will update to "Pending" immediately

4. Success notification will appear

Status Indicators

- **Green Badge "Verified"**: Application is verified
- **Yellow Badge "Pending"**: Application awaiting verification

Important Notes:

- Changes are immediate and permanent
 - No confirmation dialog (be careful with clicks)
 - All actions are logged for audit purposes
 - Readers cannot see action buttons
-

Bulk Operations (Admin Only)

Selecting Applications

1. Use checkboxes in the leftmost column
2. Select individual applications or use "Select All"
3. Selected count appears in bulk actions panel

Bulk Verify

1. Select desired applications
2. Click "Verify Selected" in bulk actions panel
3. Confirm the action when prompted
4. All selected applications will be marked as verified

Bulk Unverify

1. Select desired applications
2. Click "Unverify Selected" in bulk actions panel
3. Confirm the action when prompted
4. All selected applications will be marked as pending

Clear Selection

- Click "Clear Selection" to deselect all applications
- Useful when you want to start over with selections

Bulk Operations Best Practices:

- **Double-check selections** before performing bulk actions
 - **Use filters** to isolate desired applications
 - **Work in smaller batches** for large datasets
 - **Verify results** after bulk operations
-

Troubleshooting

Common Issues and Solutions

Cannot Access Login Page

Problem: Browser shows "This site can't be reached" **Solutions:**

- Ensure Node.js server is running (`node app.js`)
- Check if port 3000 is available
- Verify correct URL: `http://localhost:3000`

Login Failed

Problem: "Invalid credentials" error **Solutions:**

- Check username and password spelling
- Verify user exists in `login_credentials` table
- Ensure database connection is working

Database Connection Error

Problem: Server won't start or crashes **Solutions:**

- Verify MySQL server is running
- Check database configuration in `app.js`
- Ensure database and tables exist
- Verify network connectivity to database

Empty Data Table

Problem: No applications showing **Solutions:**

- Check if data exists in `installment_applications` table

- Clear all filters and try again
- Verify database permissions
- Check browser console for JavaScript errors

Role Permissions Not Working

Problem: Buttons not appearing/working for Editor/Admin **Solutions:**

- Logout and login again
- Check `role_access` field in database
- Clear browser cookies/session data
- Verify role assignment is correct

Slow Performance

Problem: Page loads slowly or times out **Solutions:**

- Reduce page size (lower limit in pagination)
- Add database indexes on frequently searched fields
- Use more specific filters to reduce dataset
- Check database server performance

Error Messages Guide

"Authentication required"

- **Meaning:** Session expired or not logged in
- **Solution:** Login again

"Insufficient permissions"

- **Meaning:** User role doesn't allow this action
- **Solution:** Contact admin for role upgrade

"Application not found"

- **Meaning:** Trying to update non-existent record
- **Solution:** Refresh page and try again

"Internal server error"

- **Meaning:** Database or server issue

- **Solution:** Check server logs and database connection

Getting Help

Log Files

- Check browser console (F12) for JavaScript errors
- Monitor server console for backend errors
- Review database logs for connection issues

Database Verification

```
sql

-- Check if tables exist
SHOW TABLES;

-- Verify user accounts
SELECT * FROM login_credentials;

-- Check application data
SELECT COUNT(*) FROM installment_applications;
```

Contact Information

- For technical issues: Contact your system administrator
- For access problems: Contact your database administrator
- For user training: Refer to this guide or contact support

Security Best Practices

For Administrators:

- Regularly update user passwords
- Monitor user activity logs
- Keep database credentials secure
- Regular backup of data
- Update Node.js and dependencies

For Users:

- Don't share login credentials
 - Always logout when finished
 - Report suspicious activity
 - Use strong passwords
 - Don't leave sessions unattended
-

Maintenance

Regular Tasks:

- **Database Backup:** Weekly full backups recommended
- **Log Rotation:** Clear old session and error logs
- **Performance Monitoring:** Check response times
- **Security Updates:** Keep dependencies updated
- **User Management:** Review and update user access

Monthly Reviews:

- Verify data integrity
 - Check system performance metrics
 - Review user access permissions
 - Update documentation as needed
 - Plan for capacity growth
-

This guide covers the essential operations of the IMS system. For additional features or customization requests, contact your development team.