

Student Billing System - Database Setup Guide

Login Credentials

All users in the system have the same password for testing purposes.

Test Accounts

Email	Password	Name	Role
test@test.com	test123	Test User	Student
john.doe@iiitb.com	test123	John Doe	Student
jane.smith@iiitb.com	test123	Jane Smith	Student
alice.j@iiitb.com	test123	Alice Johnson	Student
bob.wilson@iiitb.com	test123	Bob Wilson	Student
eva.brown@iiitb.com	test123	Eva Brown	Student

Note: All passwords are encrypted using BCrypt with strength 10.

Database Configuration

Database Name: MiniProject

Database Type: MySQL 8.0

Default Credentials:

- Username: root
- Password: Mysql123.

JDBC URL:

```
jdbc:mysql://localhost:3306/MiniProject?  
createDatabaseIfNotExist=true&useSSL=false&serverTimezone=UTC&allowPublicKeyRetrie  
val=true
```

Complete SQL Insert Statements

Create Database (if needed)

```
CREATE DATABASE IF NOT EXISTS MiniProject;  
USE MiniProject;
```

Insert Bills

```
INSERT INTO bills (id, amount, bill_date, deadline, description) VALUES
(1, 5000.00, '2024-01-01', '2024-01-15', 'Spring 2024 Tuition Fee'),
(2, 100.00, '2024-01-01', '2024-01-20', 'Library Fee 2024'),
(3, 300.00, '2024-01-01', '2024-01-25', 'Laboratory Fee'),
(4, 150.00, '2024-01-01', '2024-01-30', 'Student Activity Fee'),
(5, 800.00, '2024-01-01', '2024-01-15', 'Health Insurance Fee');
```

Insert Students

Password for all students: test123

BCrypt Hash: \$2a\$10\$stI6eRm3SaICEHRsRPIIn/eznPi3FlTzfz88FtiVrANPFicOVTftau

```
INSERT INTO students (id, cgpa, domain, email, first_name, graduation_year,
last_name, password, photograph_path, placement_id, roll_number, specialisation,
total_credits) VALUES
(1, 3.8, 'Computer Science', 'test@test.com', 'Test', 2024, 'User',
'$2a$10$stI6eRm3SaICEHRsRPIIn/eznPi3FlTzfz88FtiVrANPFicOVTftau',
'/photos/test.jpg', 100, 'R2023000', 'Testing', 100),
(2, 3.8, 'Computer Science', 'john.doe@iiitb.com', 'John', 2024, 'Doe',
'$2a$10$stI6eRm3SaICEHRsRPIIn/eznPi3FlTzfz88FtiVrANPFicOVTftau',
'/photos/john.jpg', 101, 'R2023001', 'AI/ML', 85.5),
(3, 3.9, 'Computer Science', 'jane.smith@iiitb.com', 'Jane', 2024, 'Smith',
'$2a$10$stI6eRm3SaICEHRsRPIIn/eznPi3FlTzfz88FtiVrANPFicOVTftau',
'/photos/jane.jpg', 102, 'R2023002', 'Cybersecurity', 90),
(4, 3.7, 'Information Technology', 'alice.j@iiitb.com', 'Alice', 2024, 'Johnson',
'$2a$10$stI6eRm3SaICEHRsRPIIn/eznPi3FlTzfz88FtiVrANPFicOVTftau',
'/photos/alice.jpg', 103, 'R2023003', 'Web Development', 82.5),
(5, 3.6, 'Computer Science', 'bob.wilson@iiitb.com', 'Bob', 2024, 'Wilson',
'$2a$10$stI6eRm3SaICEHRsRPIIn/eznPi3FlTzfz88FtiVrANPFicOVTftau', '/photos/bob.jpg',
104, 'R2023004', 'Data Science', 78),
(6, 3.9, 'Information Technology', 'eva.brown@iiitb.com', 'Eva', 2024, 'Brown',
'$2a$10$stI6eRm3SaICEHRsRPIIn/eznPi3FlTzfz88FtiVrANPFicOVTftau', '/photos/eva.jpg',
105, 'R2023005', 'Cloud Computing', 88.5);
```

Insert Student Bills (Relationships)

```
INSERT INTO student_bills (id, bill_id, student_id) VALUES
(1, 1, 1), -- Test User assigned Tuition Fee
(2, 2, 1), -- Test User assigned Library Fee
(3, 1, 2), -- John Doe assigned Tuition Fee
(4, 1, 3), -- Jane Smith assigned Tuition Fee
(5, 1, 4), -- Alice Johnson assigned Tuition Fee
(6, 1, 5), -- Bob Wilson assigned Tuition Fee
(7, 1, 6), -- Eva Brown assigned Tuition Fee
(8, 3, 2), -- John Doe assigned Laboratory Fee
(9, 4, 3), -- Jane Smith assigned Activity Fee
```

```
(10, 5, 4), -- Alice Johnson assigned Insurance Fee
(11, 2, 5); -- Bob Wilson assigned Library Fee
```

Insert Student Payments

```
INSERT INTO student_payment (id, amount, description, payment_date, bill_id,
student_id) VALUES
(10001, 100.00, 'Library Fee Payment', '2024-01-10', 2, 1),
(10002, 2500.00, 'Partial Tuition Payment', '2024-01-12', 1, 2),
(10003, 5000.00, 'Full Tuition Payment', '2024-01-05', 1, 3),
(10004, 5000.00, 'Full Tuition Payment', '2024-01-08', 1, 5),
(10005, 150.00, 'Activity Fee Payment', '2024-01-15', 4, 2),
(10006, 2000.00, 'Partial Tuition Payment', '2024-01-14', 1, 1),
(10007, 1000.00, 'Partial Tuition Payment', '2024-01-13', 1, 4),
(10008, 800.00, 'Insurance Fee Payment', '2024-01-16', 5, 3),
(10009, 300.00, 'Lab Fee Payment', '2024-01-11', 3, 5),
(10010, 100.00, 'Library Fee Payment', '2024-01-09', 2, 2);
```

Student Details Summary

Student 1: Test User (test@test.com)

- **Roll Number:** R2023000
- **CGPA:** 3.8
- **Domain:** Computer Science
- **Specialization:** Testing
- **Total Credits:** 100
- **Assigned Bills:** Tuition Fee (\$5000), Library Fee (\$100)
- **Payments Made:** \$100 (Library), \$2000 (Partial Tuition)
- **Outstanding:** \$3000

Student 2: John Doe (john.doe@iiitb.com)

- **Roll Number:** R2023001
- **CGPA:** 3.8
- **Domain:** Computer Science
- **Specialization:** AI/ML
- **Total Credits:** 85.5
- **Assigned Bills:** Tuition Fee (\$5000), Laboratory Fee (\$300)
- **Payments Made:** \$2500 (Partial Tuition), \$150 (Activity), \$100 (Library)
- **Outstanding:** \$2650

Student 3: Jane Smith (jane.smith@iiitb.com)

- **Roll Number:** R2023002
- **CGPA:** 3.9

- **Domain:** Computer Science
- **Specialization:** Cybersecurity
- **Total Credits:** 90
- **Assigned Bills:** Tuition Fee (\$5000), Student Activity Fee (\$150)
- **Payments Made:** \$5000 (Full Tuition), \$800 (Insurance)
- **Outstanding:** \$150

Student 4: Alice Johnson (alice.j@iiitb.com)

- **Roll Number:** R2023003
- **CGPA:** 3.7
- **Domain:** Information Technology
- **Specialization:** Web Development
- **Total Credits:** 82.5
- **Assigned Bills:** Tuition Fee (\$5000), Health Insurance Fee (\$800)
- **Payments Made:** \$1000 (Partial Tuition)
- **Outstanding:** \$4800

Student 5: Bob Wilson (bob.wilson@iiitb.com)

- **Roll Number:** R2023004
- **CGPA:** 3.6
- **Domain:** Computer Science
- **Specialization:** Data Science
- **Total Credits:** 78
- **Assigned Bills:** Tuition Fee (\$5000), Library Fee (\$100)
- **Payments Made:** \$5000 (Full Tuition), \$300 (Lab Fee)
- **Outstanding:** \$100

Student 6: Eva Brown (eva.brown@iiitb.com)

- **Roll Number:** R2023005
- **CGPA:** 3.9
- **Domain:** Information Technology
- **Specialization:** Cloud Computing
- **Total Credits:** 88.5
- **Assigned Bills:** Tuition Fee (\$5000)
- **Payments Made:** None
- **Outstanding:** \$5000

Quick Setup Instructions

Option 1: Automatic Setup (Recommended)

The application automatically creates tables and loads data on startup when using the provided `application.properties` configuration.

1. Ensure MySQL is running

2. Create database: `CREATE DATABASE IF NOT EXISTS MiniProject;`
3. Run the Spring Boot application
4. Tables and data will be created automatically

Option 2: Manual Setup

If you want to manually populate the database:

1. Create the database:

```
CREATE DATABASE IF NOT EXISTS MiniProject;  
USE MiniProject;
```

2. Let Hibernate create the tables (run the app once), OR create them manually using the schema from the logs
3. Run all the INSERT statements above in order:
 - Insert Bills
 - Insert Students
 - Insert Student Bills
 - Insert Student Payments

Testing the Application

Backend API Test (using curl or PowerShell)

Login Request:

```
Invoke-RestMethod -Uri "http://localhost:8080/api/auth/login" -Method POST -Body  
'{"email":"test@test.com","password":"test123"}' -ContentType "application/json"
```

Expected Response:

```
{  
  "message": "Login successful",  
  "status": "success",  
  "token": "eyJhbGciOiJIUzI1NiJ9..."  
}
```

Frontend Access

- **URL:** http://localhost:3000
- **Login with:** test@test.com / test123

Notes for Your Friend

1. **All passwords are the same:** `test123` for easy testing
2. **Password is encrypted:** The BCrypt hash in the database is already correct
3. **Database will auto-create:** Just ensure MySQL is running with the credentials above
4. **No manual table creation needed:** Hibernate creates all tables automatically
5. **Data loads on startup:** The `data.sql` file in `src/main/resources` contains all INSERT statements

Important Configuration

Make sure your `application.properties` has:

```
spring.datasource.url=jdbc:mysql://localhost:3306/MiniProject?
createDatabaseIfNotExist=true&useSSL=false&serverTimezone=UTC&allowPublicKeyRetrie
val=true
spring.datasource.username=root
spring.datasource.password=Mysql123.
spring.jpa.hibernate.ddl-auto=create
spring.jpa.defer-datasource-initialization=true
```

Troubleshooting

If login fails:

- Verify MySQL is running on port 3306
- Check that the password hash matches exactly (no truncation)
- Ensure BCrypt is properly configured in `SecurityConfig.java`

If tables are not created:

- Check that `spring.jpa.hibernate.ddl-auto=create` is set
- Verify MySQL connection credentials

If data is not loaded:

- Ensure `spring.jpa.defer-datasource-initialization=true` is set
- Check that `data.sql` is in `src/main/resources/`

Generated on: November 24, 2025

Application: Student Billing System

Tech Stack: Spring Boot 3.3.5 + React 18.3.1 + MySQL 8.0