

User Class Changes

User.h

In the header class, the following changes have been done.

1. Added the details function declaration
2. changed the Default parameter for the User Class. Now, only the Gender, Religion and the DOB are set as default values

User.cpp

Ok, so the primary changes have been made to the SaveUsers and the LoadUsers Function. Both of them now uses the MySQL DB to both store a user into the DB as handled by the SaveUser and then Loading the Users from the DB and into a Vector using the LoadUsers Function. The codes are given below:

```
void saveUser(const User &user)
{
    QSqlQuery query;
    QString insert_user = "INSERT INTO users (name, email, phoneNo) VALUES
('"+QString::fromStdString(user.getName())+"',
 '"+QString::fromStdString(user.getEmail())+"',
 '"+QString::fromStdString(user.getPhoneNo())+"' )";
    MySQL_Insert(insert_user);
}

vector<User> loadUsers() {
    vector<User> users;
    QSqlQuery query;

    // Execute the query to fetch all users
    if (query.exec("SELECT id, name, email, phoneNo, gender, religion, DOB FROM
users")) {
        while (query.next()) {
            string id = query.value(0).toString().toStdString();
            string name = query.value(1).toString().toStdString();
            string email = query.value(2).toString().toStdString();
            string phoneNo = query.value(3).toString().toStdString();
            Gender gender = static_cast<Gender>(query.value(4).toInt()); // Convert
int to Gender enum
            Religion religion = static_cast<Religion>(query.value(5).toInt()); //
Convert int to Religion enum
            string DOB = query.value(6).toString().toStdString();

            // Create a User object and add it to the vector
            users.push_back(User(id, name, email, phoneNo, gender, religion, DOB));
        }
        qInfo() << "✅ Users loaded successfully from MySQL!";
    } else {
        qInfo() << "❌ Error fetching users:" << query.lastError().text();
    }

    return users;
}
```

Main.cpp + MySQLUtilities Changes

The main changes Taken place is the adding of the test cases to test my User Class. Moreover, *inline* has been added to all of the MySQL Utilities Header File to prevent multiple definitions

MYSQL DML

V.V.I => RUN THE DML STATEMENTS OF MYSQL BEFORE DOING ANYTHING

```
CREATE TABLE users (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    name VARCHAR(100) NOT NULL,  
    email VARCHAR(100) UNIQUE NOT NULL,  
    phoneNo VARCHAR(20) UNIQUE NOT NULL,  
    gender ENUM('male', 'female', 'N_A') DEFAULT 'N_A',  
    religion ENUM('Islam', 'Atheism', 'Buddhism', 'Christianity', 'Hinduism',  
'Judaism', 'Sikhism', 'Zoroastrianism', 'Other') DEFAULT 'Other',  
    DOB VARCHAR(100) DEFAULT 'NONE'  
);  
delete from users;  
ALTER TABLE users AUTO_INCREMENT = 1;
```