

Connecting Android App to MySQL Database using PHP

Girish M. Tere

Abstract—This Android application sends and receives data from the server. The server will store and fetch the data in a remote database like MySQL. In this paper, how one can store and retrieve data from a remote MySQL database using PHP 7.2 and how to access data in the Android Application is demonstrated. App demonstrates CRUD operations on a database. Server side programming is done using PHP and XAMP server is used to deploy web application.

Index Terms— CRUD operations, Activity, Intent, ListView widget

1 INTRODUCTION

FOR every commercial web application CRUD operations are important. Android App is developed which interacts with server side programs developed in PHP and PHP code is interacting with MySQL database. CRUD operations can be implemented using REST framework but here in this work simple approach is used without using any external framework. Database is created in MySQL using phpmyadmin.

```
CREATE TABLE IF NOT EXISTS `movies` (
  `movie_id` int(11) NOT NULL
  AUTO_INCREMENT,
  `movie_name` varchar(50) NOT NULL,
  `genre` varchar(20) NOT NULL,
  `year` year(4) NOT NULL,
  `rating` float NOT NULL,
  PRIMARY KEY (`movie_id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1
AUTO_INCREMENT=1 ;
```

Fig 1 describes database android created using phpmyadmin.



Fig. 1 Creation of movies table in MySQL

2. DEVELOPMENT OF SERVER SIDE PROGRAMS

The php file “db_connect.php” is used for connecting php code to MySQL.

```
<?php
define('DB_USER', "root"); // db user
define('DB_PASSWORD', ""); // db password (mention
your db password here)
define('DB_DATABASE', "android"); // database name
define('DB_SERVER', "localhost"); // db server

$con =
mysqli_connect(DB_SERVER,DB_USER,DB_PASSW
ORD,DB_DATABASE);
...
Following php files are developed:
```

1. Updating data (update_movie.php)
2. Retrieving all records (fetch_all_movies.php)
3. Fetching data from a remote database in Android.
4. Deleting record (delete_movie.php)
5. Adding record (add_movie.php)
6. Retrieve a single movie detail (get_movie_details.php)

3. DEVELOPMENT OF SERVER SIDE PROGRAMS

GUI part is developed using Android activities[4]. Project structure is shown in Fig. 2. We added Internet and Check Network Status permissions in AndroidManifest.xml

```
" />
</intent-filter>
</activity>
<activity
android:name="org.girish.mysqlconnection.MovieListin
gActivity" />
<activity
android:name="org.girish.mysqlconnection.MovieUpdat
eDeleteActivity" />
<activity
android:name="org.girish.mysqlconnection.AddMovieA
ctivity" />
</application>
</manifest>
```

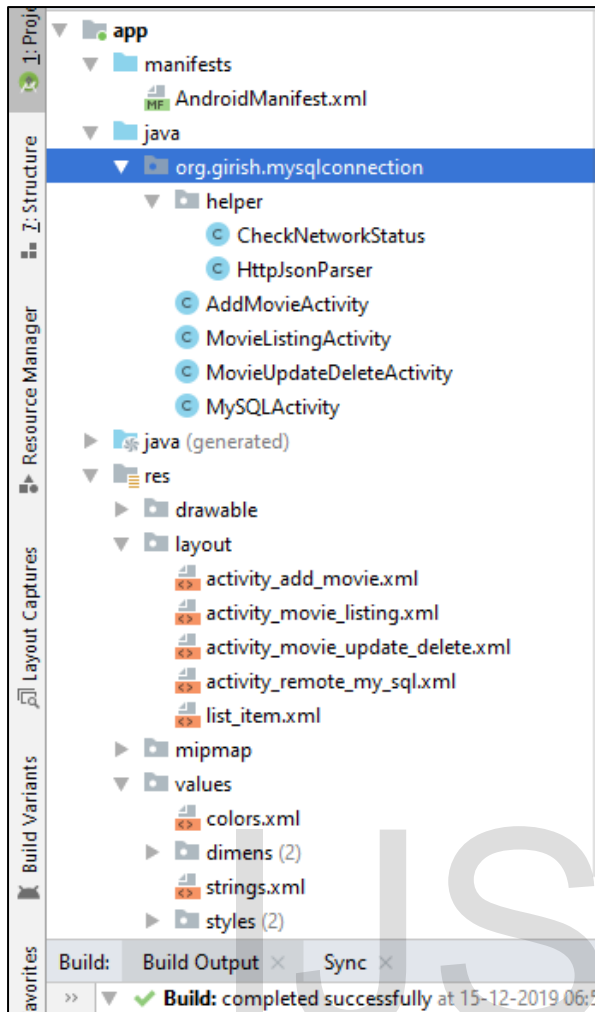


Fig. 2: MySQLConnection Application structure

Following activities are added in Android application[1]

7. MySQLActivity is Launcher activity with activity_remote_my_sql.xml layout file.
8. AddMovieActivity is used for adding new movie in database with activity_add_movie.xml layout file.
9. MovieListingActivity is used for listing all movies stored in database with activity_movie_listing.xml file.
10. MovieUpdateDeleteActivity is used for updating/deleting a movie with activity_movie_update_delete.xml layout file.

In Fig. 3 running app is shown.

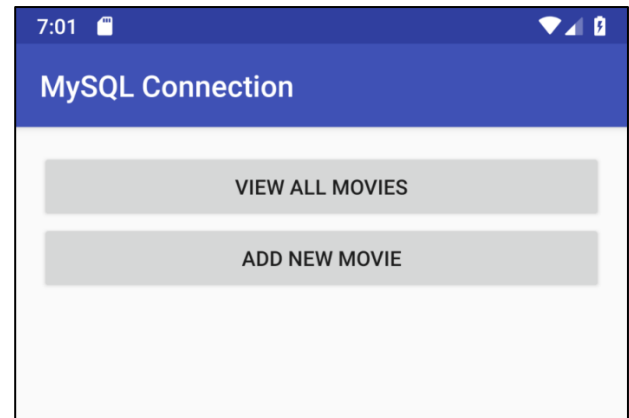


Fig. 3: First Activity (Launcher Activity)

4 MYSQL CRUD OPERATIONS

In computer programming, create, read, update, and delete (CRUD) are the four basic functions of persistent storage. Alternate words are sometimes used when defining the four basic functions of CRUD, such as retrieve instead of read, modify instead of update, or destroy instead of delete. CRUD [2] is also sometimes used to describe user interface conventions that facilitate viewing, searching, and changing information; often using computer-based forms and reports [3]. The CRUD (Create-Retrieve-Update-Delete) operations are implemented in PHP in following ways:

4.1 Adding a new Movie (Inserting a row to the database):

In file add_movie.php we check for mandatory parameters and prepare an insert query. The parameters need to be sent as POST parameter. If the movie got added successfully, then we display a success response. If there are any errors, then an error message is displayed.

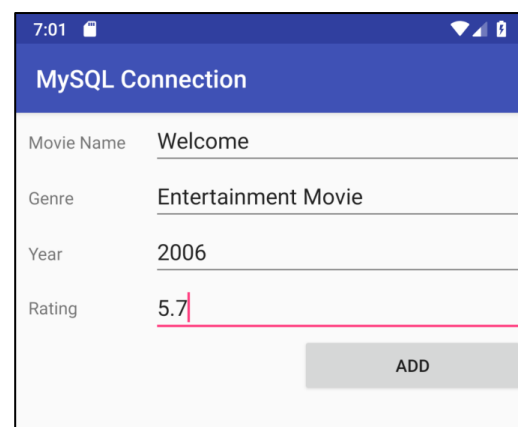


Fig. 4: Activity for adding new movie

4.2 Fetching All Movies (Retrieving data from database)

File `fetch_all_movies.php` is used for retrieving all movies. Here we query the database for `movie_id` and `movie_name` and we iterate over the fetched result and populate the result array.

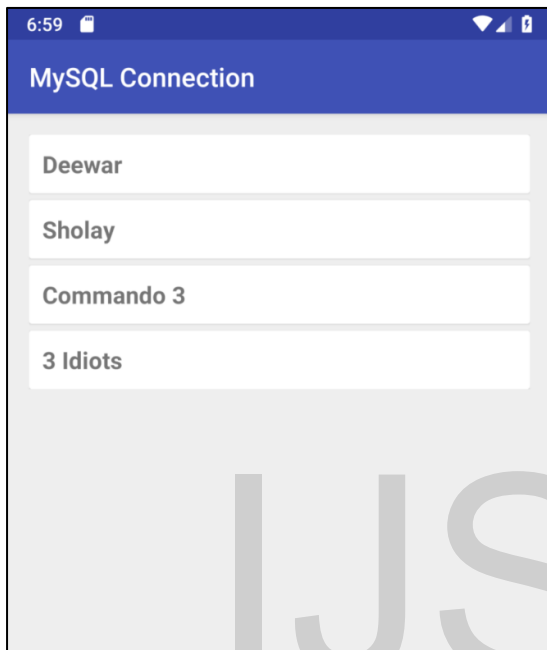


Fig. 5: Activity for listing all movies

4.3 Fetching a single movie detail (Retrieving a single row from database)

We can fetch the details of a single movie by passing the `movie_id` in the request. That is by using the primary key of the table, we can fetch a unique row from the database. We developed a file `get_movie_details.php` inside the `movies` folder and add the following code. The code expects the mandatory parameter `movie_id` to be passed in an HTTP GET method.

Movie details of a movie can be obtained using URL `http://192.168.1.14/movies/get_movie_details.php?movie_id=2`

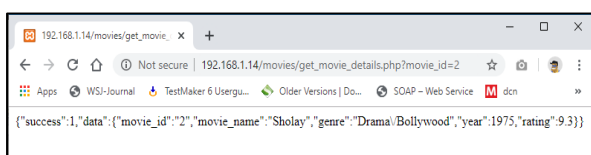


Fig. 6: Activity for listing all movies

4.4 Updating a movie (Updating a row in database)

When you wish to update a row in the database, you send the column value that needs to be updated along with the primary key of the table. Developed a file named `update_movie.php` with following code:

4.5 Deleting a movie (Deleting a row from the database)

When you want to delete a movie, you need to pass the `movie_id` in the request. Create a file `delete_movie.php` and add the following code: Here the code requires `movie_id` to be passed in an HTTP POST method. Once the movie gets deleted, you will get the following response:

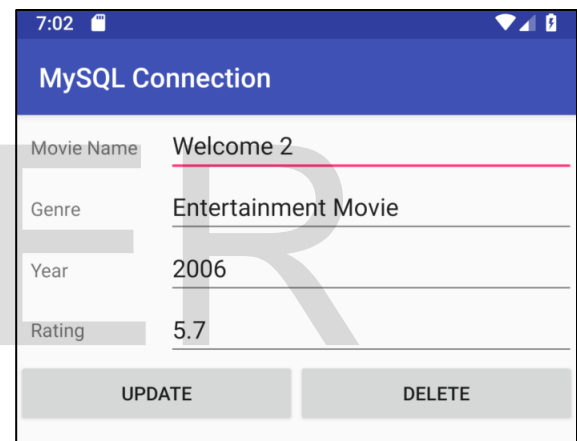


Fig. 7: Activity for updating / deleting a movie

5 CONCLUSION

MySQL Improved extension database is used for demonstration. Server side programs are developed using php 7.2 and web application is launched using Apache Web Server. Database is accessed using Android app which has necessary activities and XML layout file. Using this project we can access any remote MySQL database from mobile application and achieve interoperability.

ACKNOWLEDGMENT

Author thanks to Management of Thakur College of Science & Commerce, Mumbai for providing excellent computer lab and Principal of same college for motivating us for this research work.

REFERENCES

- [1] *Android Documentation for app developers* (2019), <https://developer.android.com/docs>, Accessed on 10th Dec 2019
- [2] *MySQL Documentation*<https://dev.mysql.com/doc/>, Accessed on 10th Dec 2019
- [3] Anupam Chugh (2019), *Android Login and Registration With PHP MySQL*, *JournalDev* <https://www.journaldev.com/12607/android-login-registration-php-mysql>
- [4] Joel Murach & Ray Harris (2014), *Murach's PHP and MySQL*, Mike Murach & Associates Inc., Third Edition,

IJSER