

# Department of Computer Sciences

## Lab Journal 11

(Fall 2018)

---

Course: **Mobile Application Development**

Date:

Course Code: CSL-341

Max Marks: 20

Faculty's Name: Miss Iram

Lab Engineer: M. Abdul Mannan

---

Name: \_\_\_\_\_ Enroll No: \_\_\_\_\_

---

### Objective(s):

To get student understand basic concept of Server Side data manipulation and its usage

### Lab Tasks:

**Task 1:** Create an activity that takes user name and roll number as an input

**Task 2:** Create a button to show data

**Task 3:** Data from server must be shown in the form of list view

### Lab Grading Sheet :

Task	Max Marks	Obtained Mark	Comments( <i>if any</i> )
1.	6		
2.	7		
3.	7		
<b>Total</b>	<b>20</b>		<b>Signature</b>

Note : Attempt all tasks and get them checked by your Lab Instructor

# Lab – Android Development Environment

---

*To get student understand basic concept of Server Side data manipulation and its usage*

A URL Connection with support for HTTP-specific features. See [the spec](#) for details.

Uses of this class follow a pattern:

Obtain a new `URLConnection` by calling [URL.openConnection\(\)](#) and casting the result to `URLConnection`.

Prepare the request. The primary property of a request is its URI. Request headers may also include metadata such as credentials, preferred content types, and session cookies.

Optionally upload a request body. Instances must be configured with [setDoOutput\(true\)](#) if they include a request body. Transmit data by writing to the stream returned by [URLConnection.getOutputStream\(\)](#).

Read the response. Response headers typically include metadata such as the response body's content type and length, modified dates and session cookies. The response body may be read from the stream returned by [URLConnection.getInputStream\(\)](#). If the response has no body, that method returns an empty stream.

Disconnect. Once the response body has been read, the `URLConnection` should be closed by calling [disconnect\(\)](#). Disconnecting releases the resources held by a connection so they may be closed or reused.

`URLConnection` will follow up to five HTTP redirects. It will follow redirects from one origin server to another. This implementation doesn't follow redirects from HTTPS to HTTP or vice versa.

If the HTTP response indicates that an error occurred, [URLConnection.getInputStream\(\)](#) will throw an `IOException`. Use [getErrorStream\(\)](#) to read the error response. The headers can be read in the normal way using [URLConnection.getHeaderFields\(\)](#),

Reference Code and Material

<https://developer.android.com/reference/java/net/URLConnection>