

---

# Baraza *Set -Up* Guide

---

Setting up Baraza Framework

---

DewCIS Solutions LTD

---

### Table of Contents

Table of Contents .....	2
Prerequisites for Baraza: .....	4
Creating a Baraza Database: .....	4
Starting a Baraza platform: .....	5
Opening an existing project: .....	6
Editing the project: .....	6
Creating a New Project: .....	7
Designing Tables: .....	8
Running Baraza Application: .....	11
Deleting a Project: .....	13
Reports on Baraza: .....	13
Making Reports: .....	13
Making the project web based: .....	18
Install Tomcat server: .....	18
Enabling connection to the database .....	19
Running the applications: .....	20

### Understanding Baraza Framework

#### What is Baraza:

The Baraza Framework is a Java Development initiative that makes application development easy. The framework takes advantage of XML to define its components. Using the framework, desktop applications, applets can be easily deployed using the same XML document with only slight modification.

Baraza is a system that incorporates different independent projects that work independently using the same Java code block that uses the individual project XML block to produce a desired result.

Baraza is divided in three different blocks;

#### 1) The Java code block

This is java code enclosed in parenthesis { } e.g. `public static void main(String[] args)`

i.e. {  
  
    *//All the Java code found here is inside a block*  
}

#### 2) XML block

This is a reusable piece (block) of content stored as well-formed, static XML, and is particularly useful when there is a large amount of XML that must be styled and included on one or more pages of a website.

#### 3) Database block

This is the smallest amount of data that a program can request. It is a multiple of an operating system block, which is the smallest amount of data that can be retrieved from storage or memory.

Database provides standardized information about the system; actually it's a large set of structured data, and run operations on the data requested by numerous users who use the system. Currently used in Baraza is PostgreSQL which probably handles virtually all the standard SQL constructs. It is easy to administer.

The Java code block is again divided into four different blocks:

- ⤴ Applet- An applet is a small application designed to run within another application written in the Java programming language.
- ⤴ Servlet- a Java programming language class used to extend the capabilities of a server, e.g. process or store data that was submitted from an HTML form.

It provides dynamic content such as the results of a database query

It also manages state information that does not exist in the stateless HTTP protocol

- ⤴ Server- A computer or computer program that manages access to a centralized resource or service in a network.
- ⤴ IDE- (Integrated/Interactive Development Environment). It is a software application that provides comprehensive facilities to computer programmers for software development. An IDE normally consists of a source code editor, build automation tools and a debugger. This program typically provides many features for authoring, modifying, compiling, deploying and debugging software. This contrasts with software development using unrelated tools, such as vi, GCC or make.

### Features of the Baraza framework include:

- ⤴ High development speed
- ⤴ Low development cost
- ⤴ Web enabled applications
- ⤴ Very stable
- ⤴ Well defined (well ordered) design
- ⤴ Ease of operation (user friendly)
- ⤴ Compatible with pc
- ⤴ Reports, mailing lists, and forums are available

### Prerequisites for Baraza:

These include:

- i. Database server, e.g. postgres is a free and open source software. It has a proven architecture that has earned it a strong reputation for reliability, data integrity, and correctness. It runs on all major operating systems, including Linux, UNIX (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64), and Windows. It also has full support for foreign keys, joins, views, triggers, and stored procedures (in multiple languages).
- ii. Java: this is an object-oriented computer programming language that is specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere" (WORA), meaning that code that runs on one platform does not need to be recompiled to run on another.

After installation of the above, download baraza, a zipped folder from the website [www.openbaraza.org](http://www.openbaraza.org). Extract or unzip it to a desired folder. This is the folder you will work from whenever you want to run the application.

### Creating a Baraza Database:

Users will approach you with what they expect a system to achieve. This is called a user story. You decipher what they need and design a system that will achieve the user's desired results.

## Baraza Set -Up Guide

Example:

*We shall create a simple database of students registering for units in a college.*

*A college wants to register students online.*

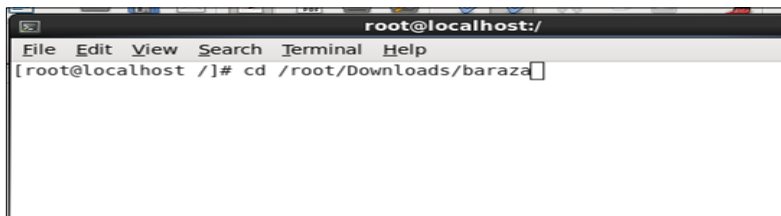
*Students once registered for the courses of their choice should be able to know which department they belong to.*

*They should also be able to know when the semester starts and ends, the name of their HOD, this means that their start and finish date should also show if in any case there aren't any unit retakes.*

*For purposes of learning, we shall start with only two tables. Ensure that you design your tables in a way that you capture all necessary data and with the right relationships.*

### Starting a Baraza platform:

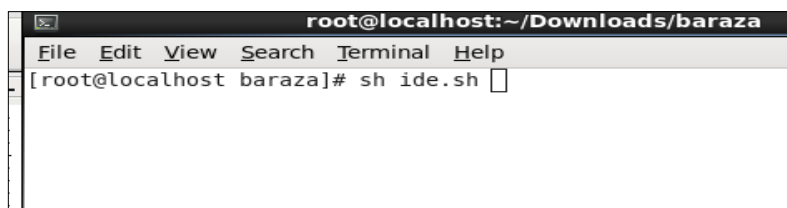
First you have to know, and always remember where you have placed your folder. Navigate to the Baraza folder you extracted. In our case, we shall have the Baraza folder in 'Downloads'



```
root@localhost:~  
File Edit View Search Terminal Help  
[root@localhost ~]# cd /root/Downloads/baraza
```

This allows you to enter into the Baraza directory.

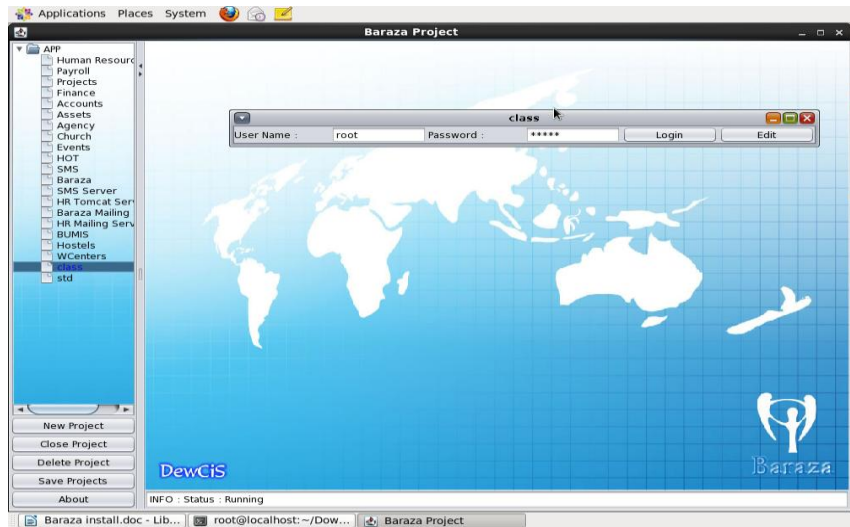
Run sh ide.sh command as `cd /root/Downloads/baraza` or `cd /./Downloads/baraza`



```
root@localhost:~/Downloads/baraza  
File Edit View Search Terminal Help  
[root@localhost baraza]# sh ide.sh
```

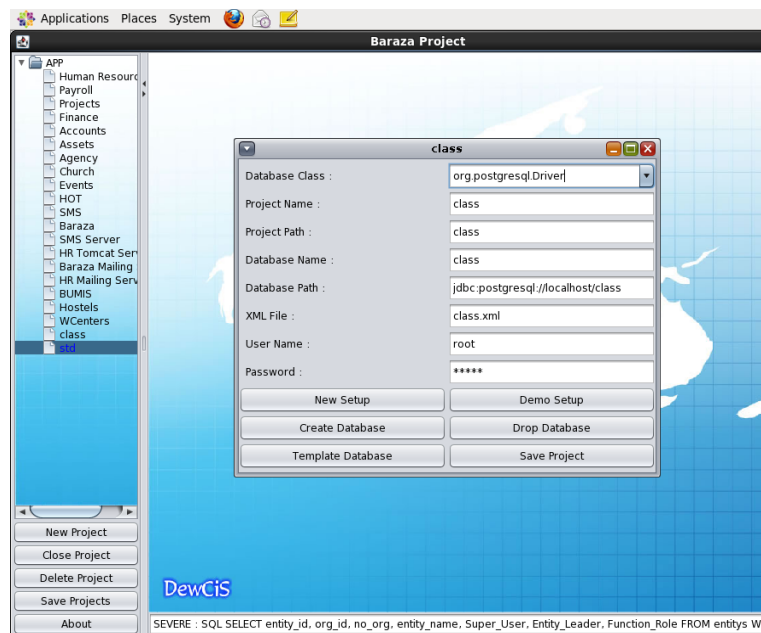
This takes you to the following window. Here, you open an existing database where you can edit or create a new database.

### Opening an existing project



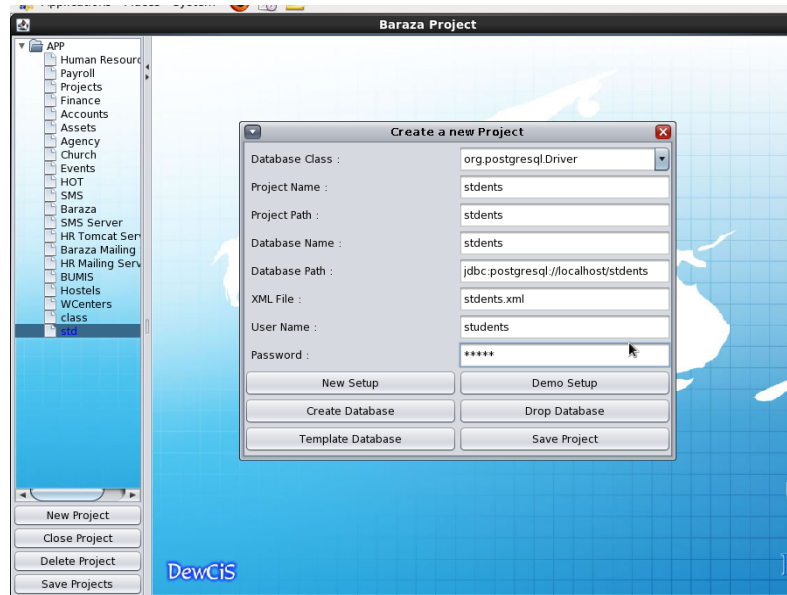
### Editing the project

When you are on the window above, click on 'Edit' at the right hand side. This gives you an option of editing your project; you can change the project name, or login passwords.



## Creating a New Project

You create a new project by clicking on the 'New project' option on the left side of the window. Give your new project your desired name and 'enter', this will generate the project path, database, name, database path, and the XML file.



On the tabs above; you can either:

Click on 'Create New' to create a new database.

'Create Demo' to create a demo database.

Click on 'Template database'. This imports data from Baraza, and auto generates XML. Save your project.

If you click on 'Create Database', you will create a blank database therefore you have to enter all data in the tables manually.

Click on 'Query builder', then database files. This gives you a window where you design your tables.

Always click on 'Save Project' to save any changes. It is important to use strong passwords to stop unauthorized users from accessing the database.

After login, you can create your tables at the 'Query Builder' 'Database Files'. Carefully give each field the right data type, and when this is done, execute then save.

## Designing Tables:

CREATE TABLE department(

dept\_id integer references students,

course\_name text,

hod\_name text

);

CREATE TABLE students (

student\_id serial primary key,

student\_name text,

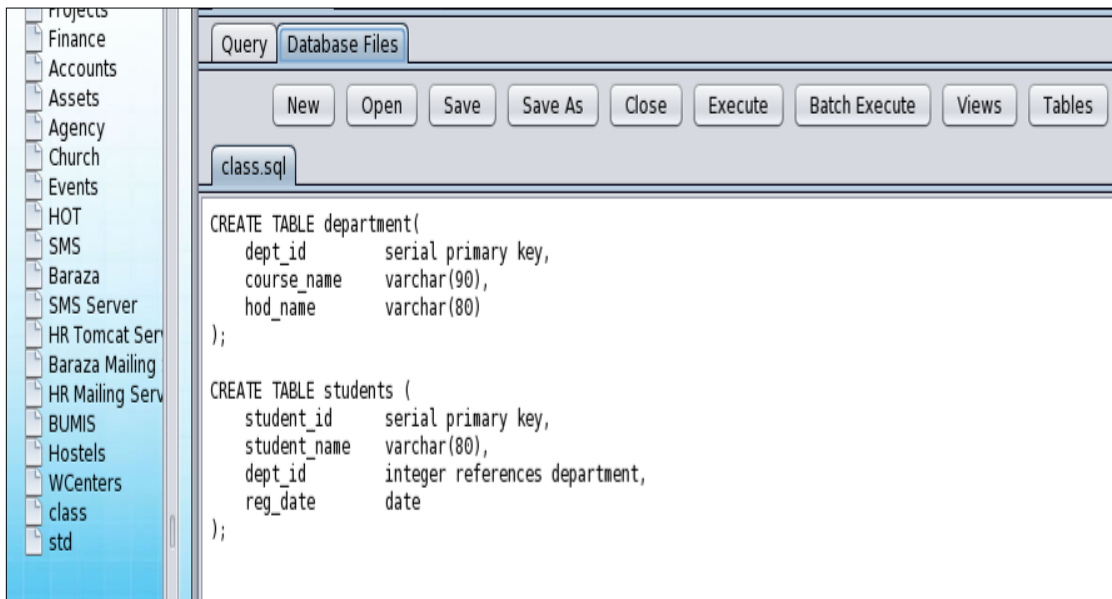
course\_name text,

dept\_id integer,

reg\_date date

);

Click on 'Database Files' 'class.sql' to get to the window where you design your tables. It looks like this:

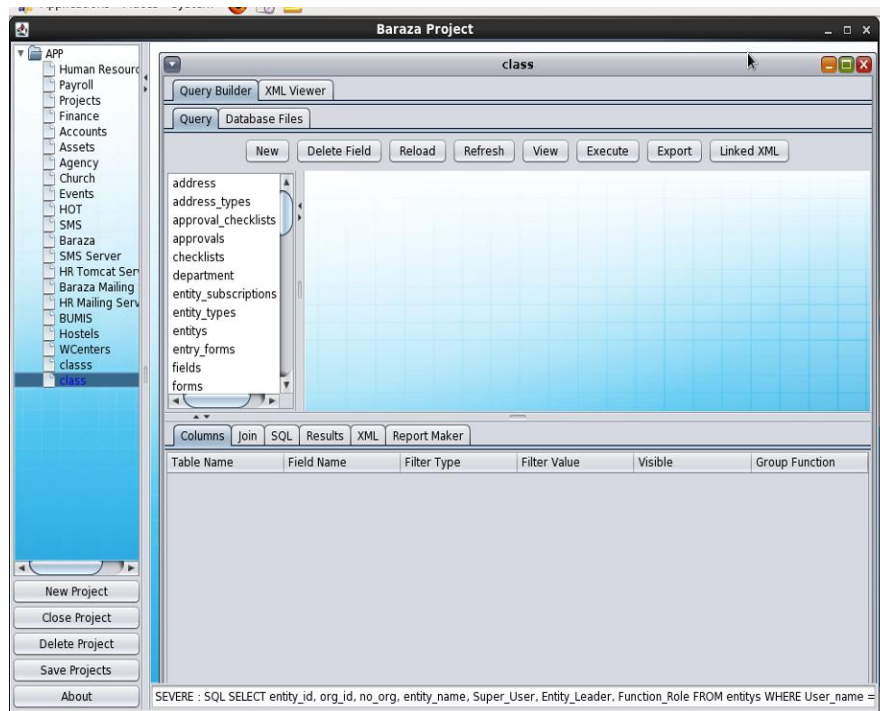




Execute and save your tables. In case there is any mistake while doing so, you will get an error at the bottom of that window. Read to correct any, then execute and save to commit any changes. When the tables are executed successfully, click XML viewer, then XML tab then New Components.

New Components allows you to create new desks and nodes on your databases, these are links of the tables and forms.

You will get a window that looks like this:

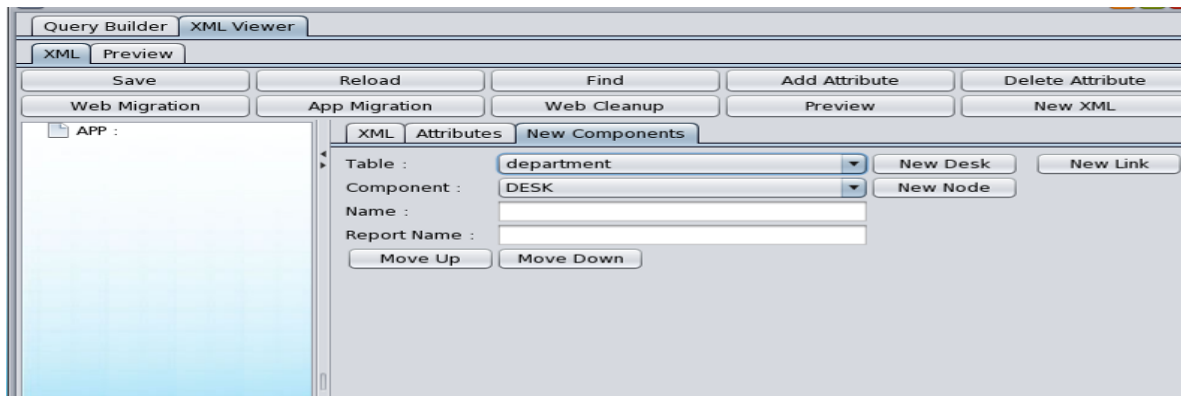


After your XML is saved, then you click on 'Query builder' then 'Query' to see where all the tables appear as above.

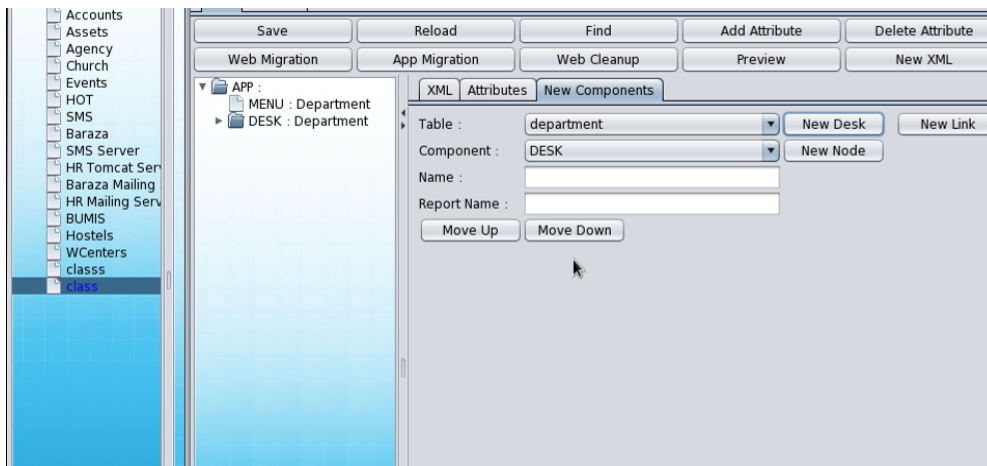
Then you create a new desk (form). A form is a window or screen that contains numerous fields, or spaces to enter data. Each field holds a field label so that any user who views the form gets an idea of its contents. A form is more user friendly than generating queries to create tables and insert data into fields.

Click on 'XML Viewer' then XML. You get to the window with a tab (New components). The new desk you add will be displayed on the left hand side of the window under 'App'.

## Baraza Set-Up Guide



You get a window that looks like this:



Save and reload to commit changes.

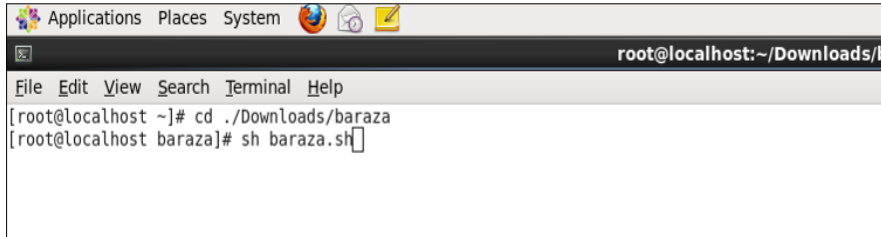
You can navigate through the tables using the 'Move Up' or 'Move Down' tabs as shown above.

Now click on XML Viewer, then XML. This gives you the XML code with all the table data. Group all table menu together at the top after 'App', and enclose it with the menu tags i.e. <MENU></MENU>.

Alternatively you can simply **cut** the MENUS that are below the main one and **paste** it to the topmost one so you have one MENU and all the DESKS are under it.

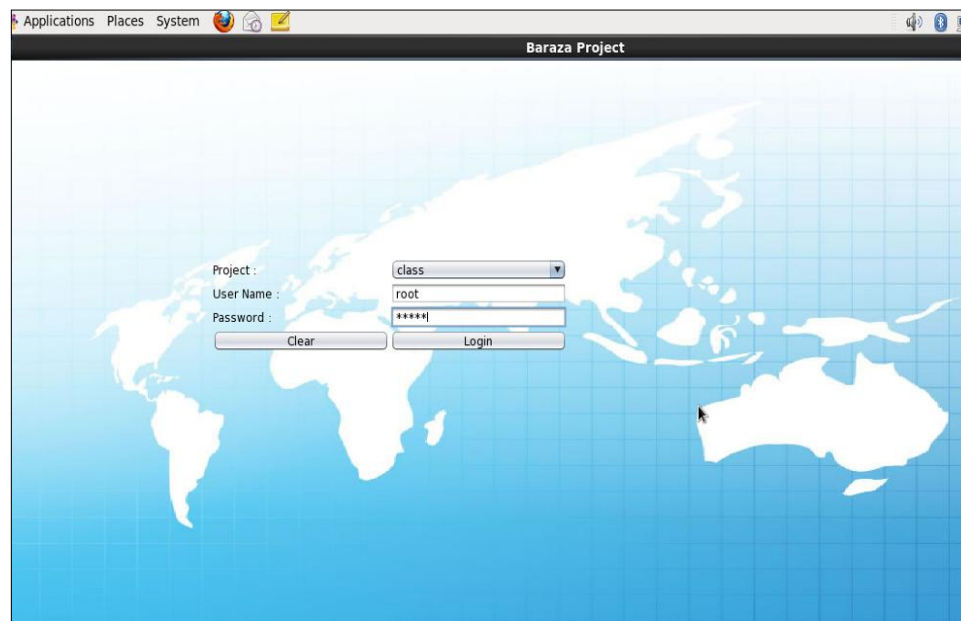
When you enter data in your forms, save all new data to commit changes.

## Running Baraza Application:

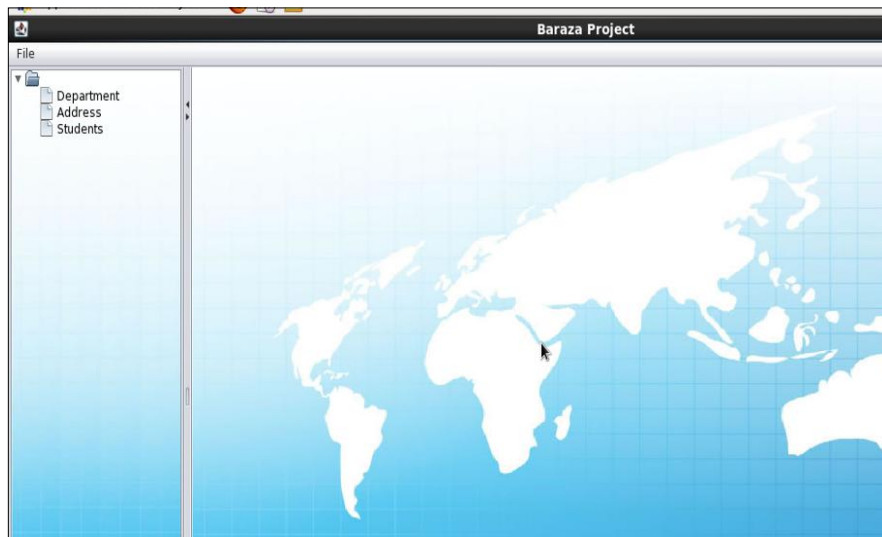


```
Applications Places System
root@localhost:~/Downloads/b
File Edit View Search Terminal Help
[root@localhost ~]# cd ./Downloads/baraza
[root@localhost baraza]# sh baraza.sh
```

Run the above command on the terminal to open Baraza. You will get a login window as shown below.

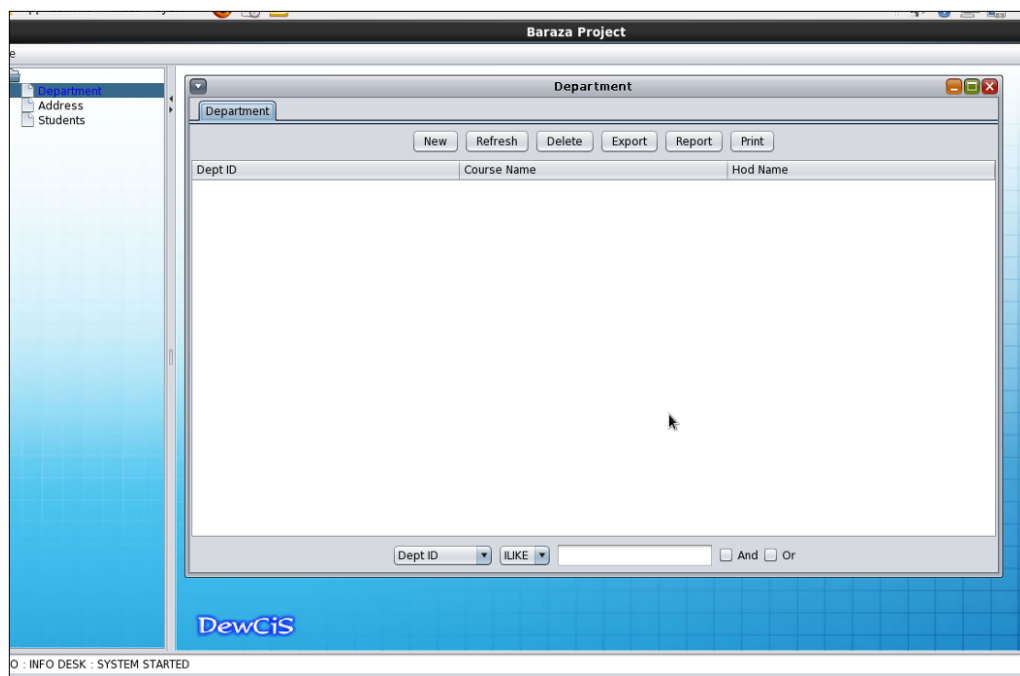


Login in using your username and password to go to the desk (forms). This displays all the desks you have created. They are dependent on the number of tables you created.



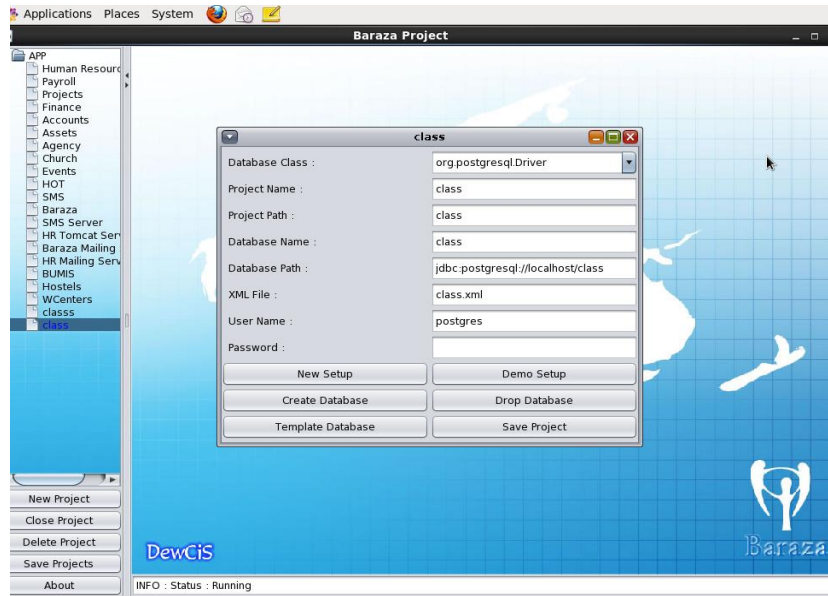
On the bottom most part you will get a message 'INFO DESK:SYSTEM STARTED'. This means you now have your desk and you can open any desk to add the data you want. This is shown on the window below.

Always save to commit any changes.



## Deleting a Project

To delete a project, you 'Drop Database' from the menu that is displayed after you click the 'edit' button.



## Reports on Baraza:

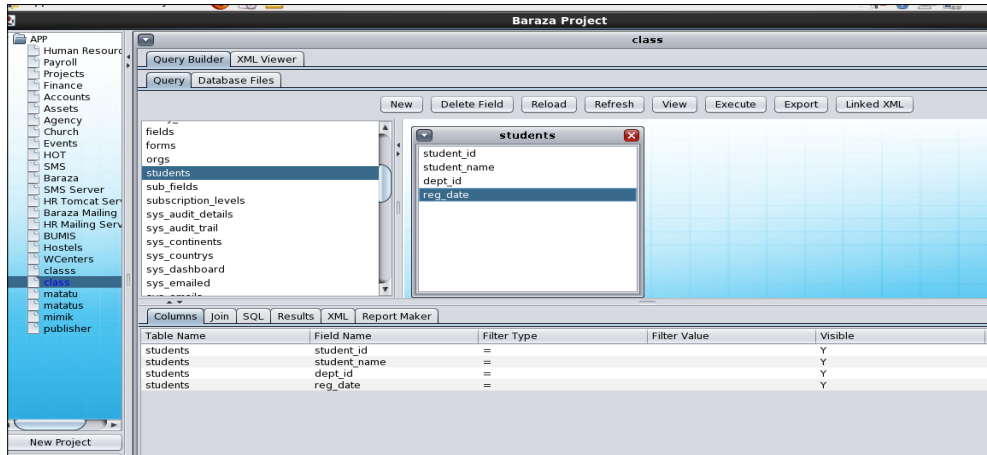
A report is a formatted result of a database query and contains useful data for decision-making and analysis. Most good business applications contain a built-in reporting tool; it is simply a front-end interface that calls or runs back-end database queries that are formatted for easy application usage.

## Making Reports:

Double click on your tables, select the fields you want to add, they will automatically be generated in columns on the bottom window. The report is generated based on the view you select here. If you'd like to change how information is displayed in the report, you must first make changes to the source view.

From our simple example of database 'class', we can select the fields of table students:

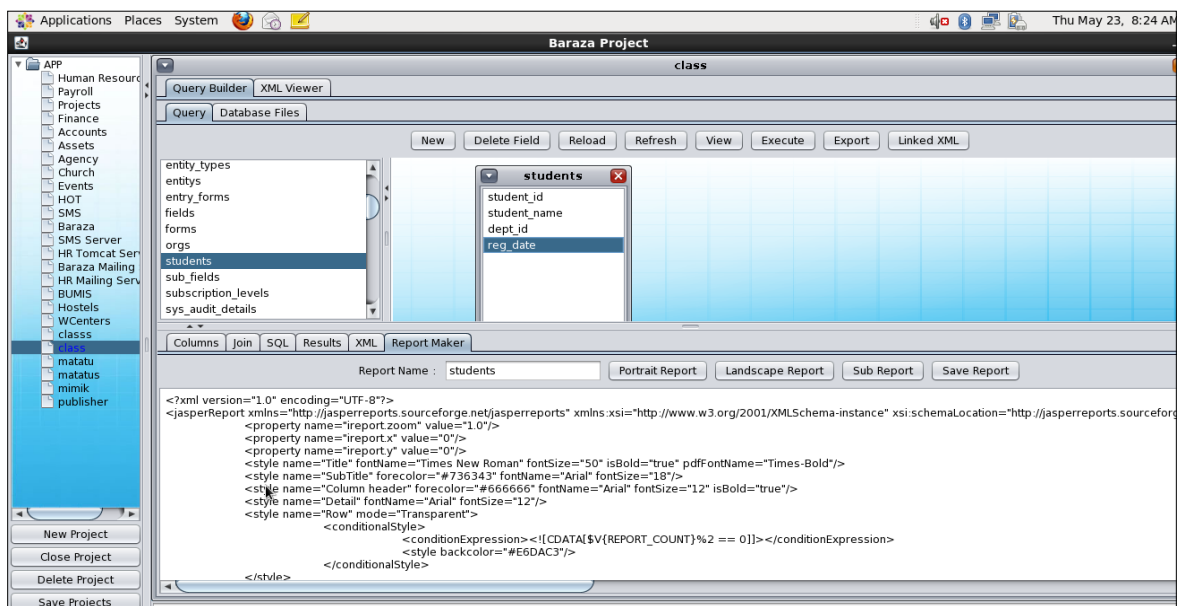
They will appear on the bottom window of your screen under 'columns'.



Select the 'Report Maker' and enter the name of the report. The XML is generated when you double click on the fields in a table, then the columns are created.

After this you choose the type of report you want to produce, e.g. Portrait, or landscape, or sub report. This will give you the XML.

The window looks like this:

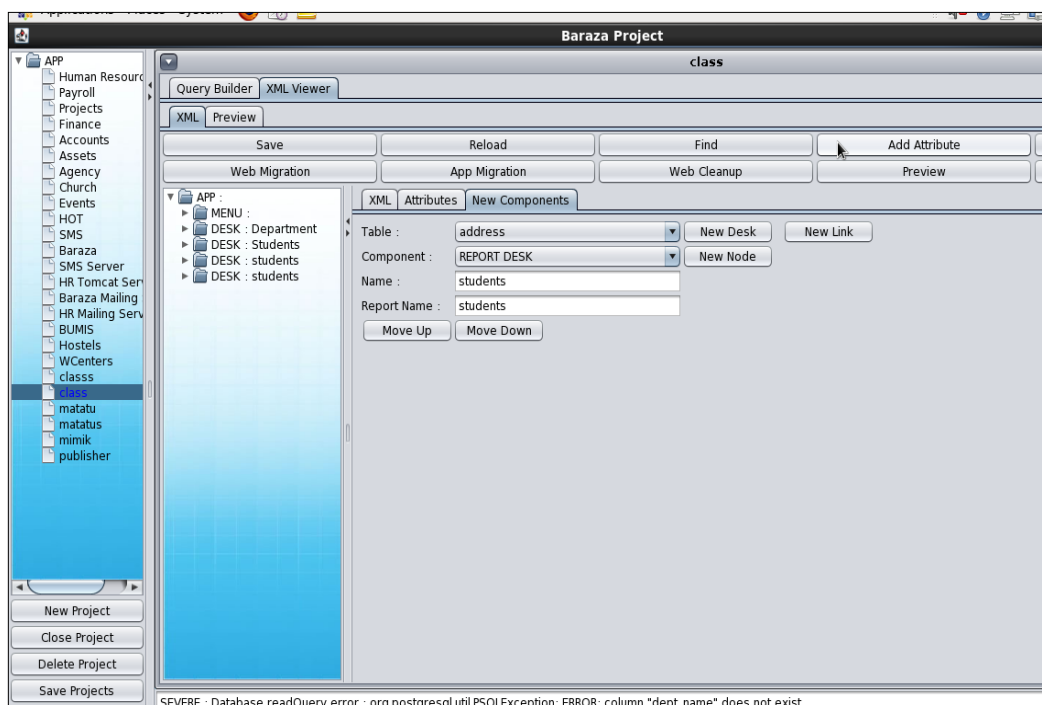


### Save the XML

Go to 'New Components' tab. On the 'Components' select 'REPORT DESK', and enter the name of the report. You can copy the report name to the 'Name' and 'Report Name' under 'New Components' tab to avoid mistakes. This ensures that the report name appears exactly as it is.

Click on 'New Node' then go to the XML tab. This will add a new desk on the left hand side window under 'APP'. Again remember to group your MENUS.

Reload to add the report to Baraza:

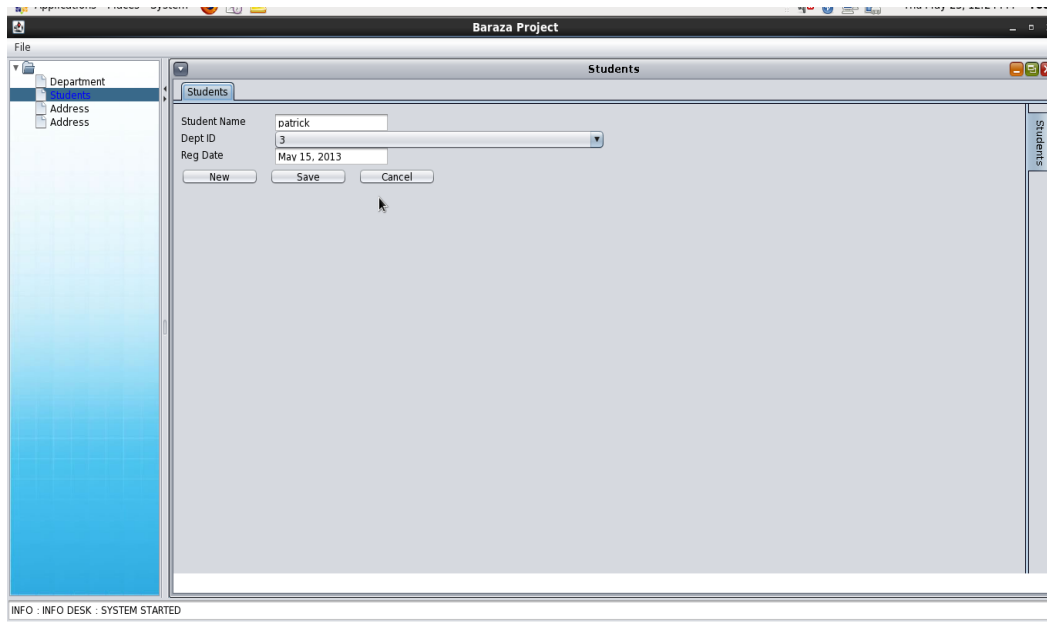


After you select 'REPORT DESK' and given the report a name, then you can create a new link. This displays your tabs on the right hand side showing the tables you have updated.

This is optional, you can display them one at a time.

Add the data you want into the forms, and save. Your report is now added.

## Baraza Set-Up Guide



Open Baraza: using sh baraza.sh





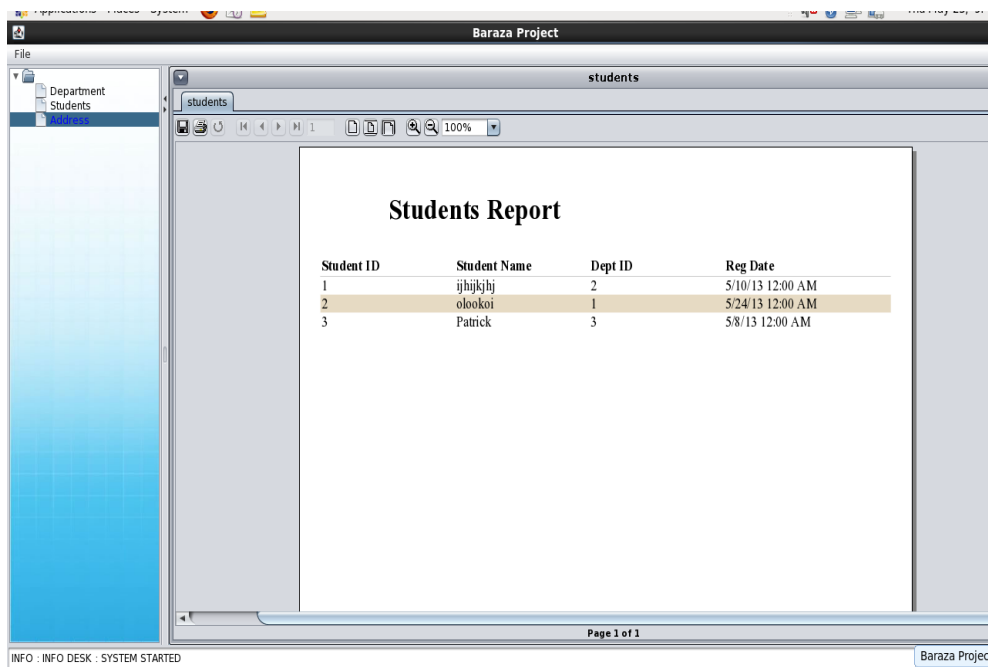
## Baraza Set -Up Guide

You get here:



Click on the report you created to open the report.

It looks like this:



## Making the project web based:

### Install Tomcat server:

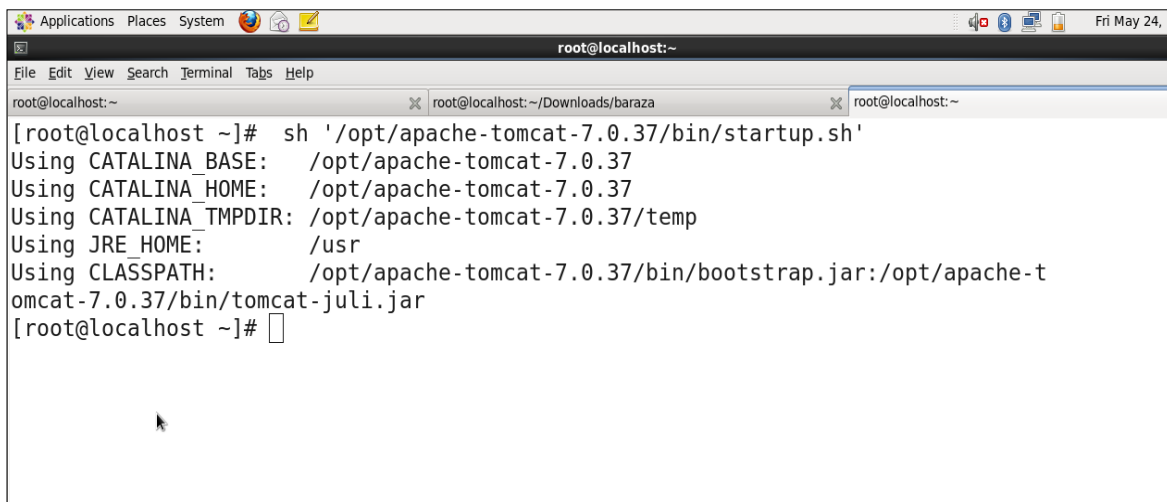
Apache Tomcat includes tools for configuration and management, but can also be configured by editing XML configuration files.

Download Apache Tomcat from the internet, and unzip it to a desired folder. Tomcat is an open source web server and servlet container that includes many additional features that make it a useful platform for developing and deploying web applications and web services.

### Starting the server:

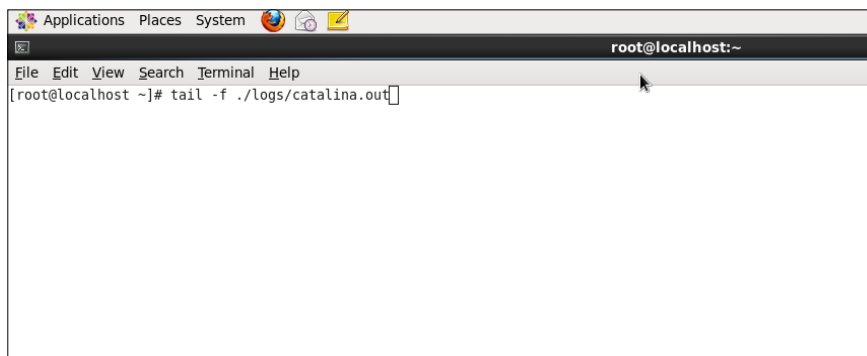
Navigate to the server directory then bin directory the run the following command.

sh startup.sh



```
root@localhost:~  
[root@localhost ~]# sh '/opt/apache-tomcat-7.0.37/bin/startup.sh'  
Using CATALINA_BASE:   /opt/apache-tomcat-7.0.37  
Using CATALINA_HOME:   /opt/apache-tomcat-7.0.37  
Using CATALINA_TMPDIR: /opt/apache-tomcat-7.0.37/temp  
Using JRE_HOME:        /usr  
Using CLASSPATH:       /opt/apache-tomcat-7.0.37/bin/bootstrap.jar:/opt/apache-t  
omcat-7.0.37/bin/tomcat-juli.jar  
[root@localhost ~]#
```

You can run your logs to see where you have saved your Tomcat file. Run the command below: this checks what is going on in the server.



```
root@localhost:~  
[root@localhost ~]# tail -f ./logs/catalina.out
```

There are several folders within Apache tomcat.

To obtain a web structure in your database, copy an XML file from an existing project to the project you create.

Copy an existing web application and paste to the tomcat server: i.e baraza.war to the 'Webapps' folder. This will automatically copy Baraza into tomcat.

Go to the folder you copied e.g hr WEB-INF folder then the configs folder and replace the hr.xml with your XML e.g class.xml

Go back to one step (to WEB-INF) folder and edit the the main xml in the section with the tag.

<context-param>

<param-name>config\_file</param-name>

<param-value>**class.xml**</param-value>

</context-param>

**N/B** Replace the bold text with the xml name of your application, e.g. We shall rename ours 'class' as per the name of our project.

## Enabling connection to the database

Go to the your web folder and edit the config.xml in the META-INF folder according to your project.

Change the database, username and the password as desired.

### Example:

<Context crossContext="true" reloadable="true">

<Resource mail.smtp.user="root" mail.from="root@localhost" name="mail/Session"  
mail.smtp.host="localhost" mail.smtp.auth="false" type="javax.mail.Session"  
password="invent" auth="Container"></Resource>

<Realm userTable="tomcat\_users" connectionURL="jdbc:postgresql://localhost/**class**"  
connectionPassword="Baraza2011" userNameCol="user\_name" roleNameCol="entity\_role"  
userCredCol="Entity\_password" digest="md5"  
className="org.apache.catalina.realm.JDBCRealm" userRoleTable="tomcat\_users"  
driverName="org.postgresql.Driver" connectionName="postgres"></Realm>

```
<Resource maxActive="150" maxIdle="50" username="postgres"
name="jdbc/database" maxWait="50" removeAbandoned="true"
driverClassName="org.postgresql.Driver" type="javax.sql.DataSource"
password="Baraza2011" url="jdbc:postgresql://localhost/class" auth="Container"
removeAbandonedTimeout="50"></Resource>
```

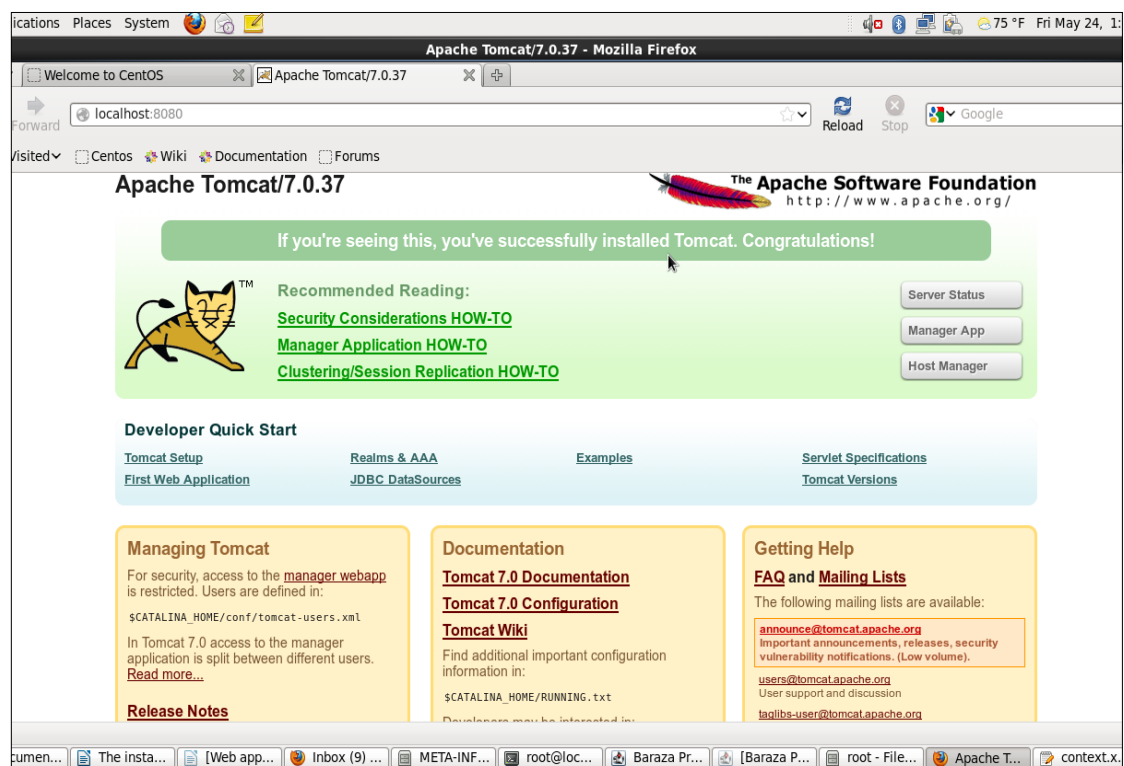
```
</Context>
```

## Running the applications:

Type the server address to your browser. e.g localhost:8080 or 192.168.0.2:8080

This brings the Apache tomcat server home page.

When you type the address above, it takes you to the Tomcat home page which looks like this:



Add your application name to access it e.g localhost:8080/class

If successful the application will bring the log in page.

Enter the credentials and continue

## Baraza Set -Up Guide

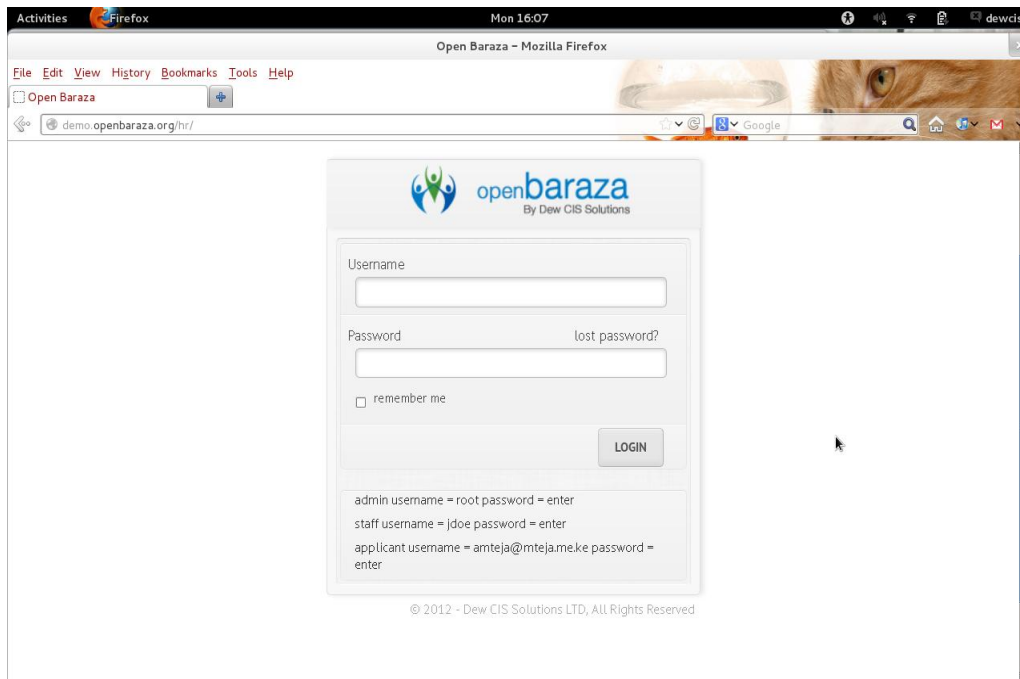
If not successful start debugging by testing the database connection but will depend on the type of error.

NB Ensure the necessary library e.g postgres,activation etc are copied in the lib directory of the tomcat folder.

If you want access the database from a different computer, then you type the IP address to the server, otherwise use the local host.

Type `www.localhost:8080` on the web browser.

You will get a window that looks like this.



This window allows you to login.

Enter your credentials and enjoy using Baraza!