



Micello's Automated Report Generator

Akshay Kumar*
Sophomore Undergraduate
Department of Computer Science & Engineering
IIT Kanpur

Project Advisor: Mr. Anil Agarwal[†]
President, Micello India
Micello Inc.

July 18, 2012

*akshay.kumar@micello.in

[†]anil@micello.com



Published by:
Micello India
506, 507
Mauryansh Elanza
Behind Parekh Hospital
Shyamal Char Rasta
Satellite
Ahmedabad

Support Phone: +91 97129 21394
+1 650 641 0712
Web: www.micello.com

Copyright Micello Inc. All rights reserved
Documentation produced July, 2012

Contents

1	Project Description	5
1.1	Brief Overview of Micello	5
1.2	My project as an Internee	5
2	Overview of the process developed	6
2.1	Designing Coverage Database	6
2.2	Generating a tex file using coverage database based on user specified filters	6
2.3	Making a pdf/dvi/ps/etc. using tex file	6
3	Coverage Database	7
3.1	Scheme of different tables	7
3.1.1	try	7
3.1.2	area	7
3.1.3	country	8
3.2	Generating different tables	8
3.3	Table try	8
3.3.1	Internal structure of stored procedure try()	9
4	User Interface with appropriate options	12
4.1	Table	13
4.2	Bar Graph	14
4.3	Maps	15
4.4	Login feature of the frontend	15
5	Backend of the report generator	16
6	Future Work	17

1 Project Description

1.1 Brief Overview of Micello

Micello is the leading provider of comprehensive indoor venue maps for mapping and navigation companies, retailers, hospital groups, mobile carriers and application developers. They create indoor venue maps and navigation content that powers location based solutions around the world. The company, founded in 2007, is privately held and headquartered in Sunnyvale, California with international offices in Japan and India. Micello has till date mapped thousands of locations throughout the world and been featured on VentureBeat, ReadWriteWeb and TechCrunch, and the company has also earned numerous accolades at various tech conferences.

The maps created by Micello are geo-coded to the real world allowing you to see the exact location as pins and get directions to other area landmarks and infrastructure like roads or parking lots. Another attractive feature is that the maps can be cached for offline usage. This comes as a great relief to frequent flyers since they will be able to scope out their route to next connection before the captain announces the use of portable gadgets is permitted again. Micello plans to integrate their map database with other sorts of information within their application.

1.2 My project as an Internee

Micello publishes a Micello Coverage Report on a monthly basis. The report includes various kinds of information pertaining to different maps published, their present status, the future endeavours of Micello and so on. Ideally speaking, the report contains various graphs of different categories, tables giving valuable information about different assets mapped by Micello, some screenshots of the maps highlighting the landmarks mapped by Micello and some other stuff. Till date, the process of report generation used to be a uphill ordeal. The report had to be generated manually individually designing all the tables, bar graphs, etc. & importing them into the report. With the help of the process developed, the time consumed in generating the report can be cut down from nearly 20 hours to a matter of few minutes.

2 Overview of the process developed

The whole of the process developed can essentially be broken down into undermentioned phases:

2.1 Designing Coverage Database

- Generate the coverage database extracting all the relevant information from micemaps41 database.
- Use mft database to generate a csv file giving values if CId along with their respective status.
- Upload/update the status in coverage database
- With the help of the jar file, calculate the individual area of the maps and store it as a text file.
- Convert the text file into a CSV or XLS
- Area is in sq. metre. So multiply it by 10 & get it in sq. ft.
- Update the coverage database with the area.
- Run a sequence of sql commands to achieve the following tasks:
 1. Convert country code to country name
 2. Fix duplicate countries like US, USA
 3. Generate Region Code viz. W, NW, NE, etc.
 4. Generate category based on the area

2.2 Generating a tex file using coverage database based on user specified filters

Coverage database has all the information needed to generate the report.

- Extract the data into relevant database tables which can be suitably embedded into the report.
- Use the user based filters to generate tables, bar graphs, maps, etc.
- Make a tex file importing all the relevant details.

2.3 Making a pdf/dvi/ps/etc. using tex file

Using the tex file generated, the report can designed in appropriate format.

- Use pdflatex for .pdf.
- latex for .dvi.
- latex for .ps.

3 Coverage Database

The first and foremost step was to develop the coverage database¹ which has all the relevant tables required for generating report. All in all, following are the relevant tables in coverage database:

try This is the most important table communicating directly with the user interface. It has all the relevant information pertaining to cid, cname, etc. etc.

area This table is generated after converting the csv area file into a table. It is useful for designating the category of the maps.

country This table contains the country name along with their code.

Note that apart from this, there are a few other tables also which are used for report generation having nothing to do with the actual repository of data.

3.1 Scheme of different tables

3.1.1 try

Field	Type	Null	Key	Default	Extra
cid	int(11)	NO	PRI	NULL	
cname	varchar(255)	YES		NULL	
street_address	varchar(255)	YES		NULL	
city	varchar(20)	YES		NULL	
state	varchar(20)	YES		NULL	
zip	int(11)	YES		NULL	
type	varchar(20)	YES		NULL	
region	varchar(20)	YES		NULL	
cat	int(11)	YES		NULL	
published	int(11)	YES		NULL	

3.1.2 area

Field	Type	Null	Key	Default	Extra
cid	int(11)	NO	PRI	NULL	
cname	varchar(255)	YES		NULL	
a0	int(11)	YES		NULL	
a1	int(11)	YES		NULL	
a2	int(11)	YES		NULL	
a3	int(11)	YES		NULL	

¹From now on, coverage database will be called reports database as another coverage database already exists

3.1.3 country

Field	Type	Null	Key	Default	Extra
code	varchar(3)	NO		NULL	
name	varchar(30)	NO		NULL	

3.2 Generating different tables

All the aforementioned tables have been generated using tables from micemap41 database which is upgraded which addition of new maps.

The table country has been directly copied from micemaps41 as its sole purpose is to store the code of different countries. As far as table area is concerned, it stores the areas of different places. Each building has four attributes *viz.* a0, a1, a2 & a3 which correspond to inner area of the building, outer area of the building, number of stores, etc. This is useful in assigning categories to the buildings.

Now comes the most important table - try: it contains all the important information of the database like Community Id (cid), Community Name (cname), Street Address (street_address), City (city), State (state), Postal Code (zip), Type of Building (type), Region (region), Category (cat) and Status of Release (published).

3.3 Table try

The table try contains all the relevant information pertaining to cids, cname, address and other such attributes. This table is dynamically generated every time you call a stored procedure named *try*. To call this stored procedure, simply type in the following sql command:

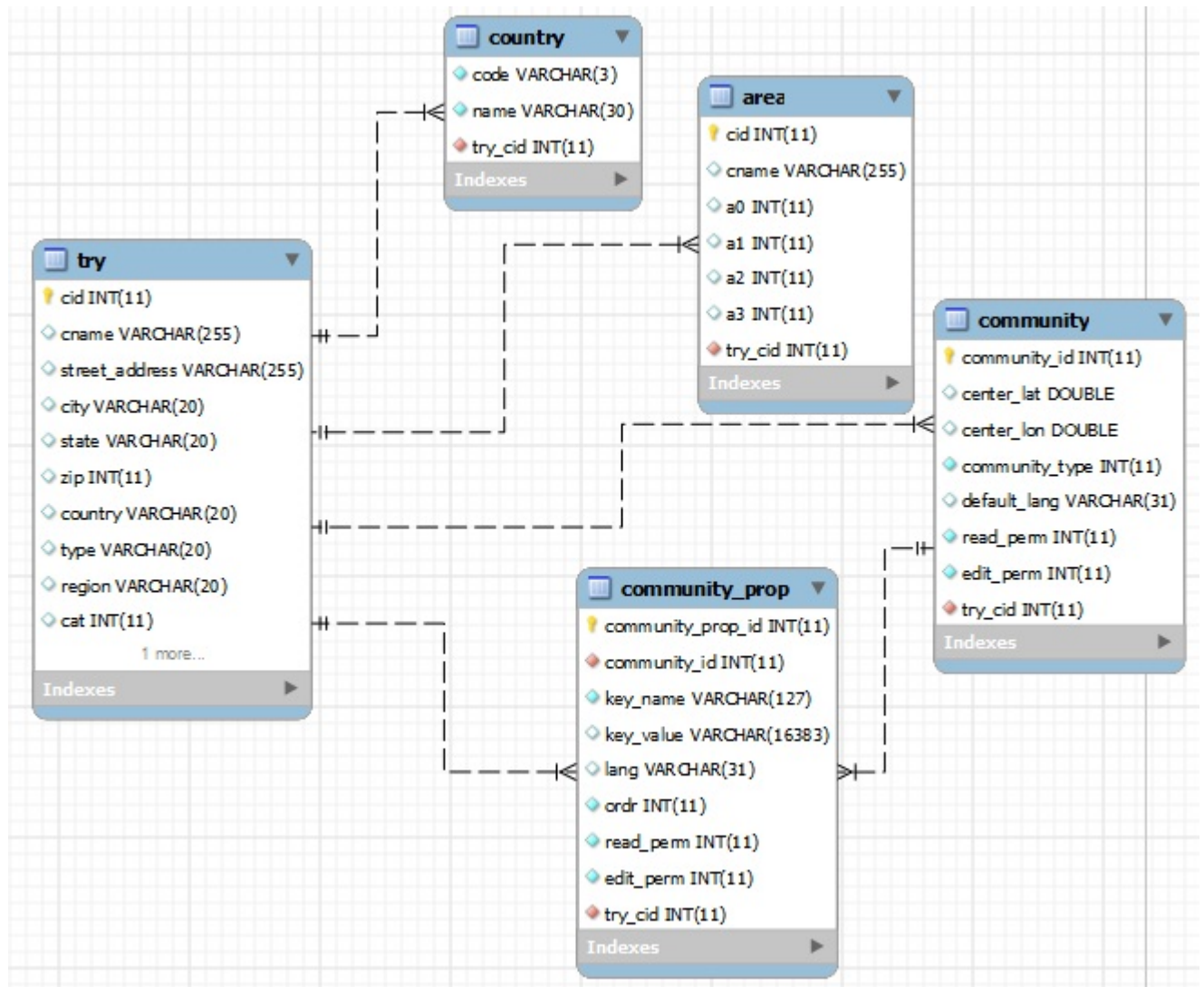
```
mysql> call try();
```

Upon successful execution, following message is returned:

```
mysql> call try();  
Query OK, 853 rows affected (11.18 sec)
```

The number of rows will vary accordingly.

3.3.1 Internal structure of stored procedure try()



1. The table `try` is dropped, if exists.
2. A fresh table `try` is created with the previously mentioned schema.
3. After creating the table, the first step is to populate it with `cids` (community id). The `cids` are copied from `micemaps41.community` table.
4. After that, other attributes such as `cname`, `street_address`, etc. are filled in. For this, the table `micemaps41.community_prop` is used. The schema of table `micemaps41.community_prop` is as follows:

Field	Type	Null	Key	Default	Extra
community_prop_id	int(11)	NO	PRI	NULL	auto_increment
community_id	int(11)	NO	MUL	NULL	
key_name	varchar(127)	NO	MUL	NULL	
key_value	varchar(16383)	YES		NULL	
lang	varchar(31)	YES		NULL	
ordr	int(11)	NO	MUL	0	
read_perm	int(11)	NO		1	
edit_perm	int(11)	NO		2	

The relevant column for us are community_id, key_name and key_value which relates the cids to different attributes of the building. The distinct key_name are

```
mysql> select distinct(key_name) from community_prop;
+-----+
| key_name |
+-----+
| city      |
| country  |
| description |
| email     |
| fax       |
| hours     |
| name      |
| phone     |
| postal code |
| province  |
| short name |
| state     |
| street address |
| url       |
+-----+
14 rows in set (0.00 sec)
```

with their corresponding key_values. As clear from the above result, the table try is updated with city, country, name, zip, state, street address, etc. from here.

5. To get the category of the buildings, the area must be calculated first. For this, a java program is run which returns a csv file with corresponding area. This csv file is converted into the table area.

The following parameters are used to desinate categories based on their areas:

- **Category 1:** Properties $\geq 1,000,000$ sq. ft.
- **Category 2:** Properties between 500,000 - 1,000,000 sq. ft.
- **Category 3:** Properties between 200,000 - 500,000 sq. ft.
- **Category 4:** Properties between 100,000 - 200,000 sq. ft.

6. To decide the type of map, following criteria is used:

- North-East covers the states of ME, VT, NH, NY, MA, RI, CT, PA, NJ, DE, MD, DC
- South-East covers the states of KY, WV, VA, TN, NC, AR, LA, MS, AL, GA, SC, FL and Puerto Rico
- Mid-West covers the states of ND, SD, NE, KS, MN, IA, MO, WI, IL, MI, IN, OH

- South-West covers the states of AZ, NM, OK, TX
 - West region covers the states of WA, MT, OR, ID, WY, CA, NV, UT, CO, AK, HI
7. In this way, category and type are decided based on simple sql queries.
 8. Lastly, the try table is updated with the status (released/published/not released) of all the maps.

4 User Interface with appropriate options

After the reports database is ready, the next step is to ask the user for various criteria and filters using which the report is to be generated. For this, a php script has been designed asking for various inputs (in the form of tables, bar graphs, and maps: more filters can be added if needed). Based on these criteria, appropriate graphs and tables are generated which are embedded into the final report.

Snapshots here shows the index.php page which contains various filter, etc. for the report. Remember to Reset once after logging into the report generator. After you have specified all the relevant criterias, click on Generate Report to generate a .tex file. You can then log out by clicking on Sign Out button.



As clear from the picture above, till now, there are three filters introduced in the reports. They are: Table, Bar Graph & Maps.

4.1 Table

Micello's Monthly Report Generator

Reset Sign Out

Create the following filters:

Table Bar Graph Maps Akshay Kumar

TABLE FORM

Community ID ☒

Community Name ☒

Street Address ☒

City ☒

State ☐

Postal code ☐

Country ☐

Type ☐

Region ☐

Category ☐

Published ☐

Title

Submit Table Form

If finished adding various features, proceed to report generation:

Generate Report

As evident from the table form, you can fill out relevant parameters to generate a table of your requirements. At the backend, the php script does a pattern globbing for the required parameters to match the criteria. The columns you want to include in the table, can be checked accordingly. By default, Community ID, Community Name, Street Address and City are checked whereas others are unchecked. They can also be checked or unchecked accordingly. After finishing this, do not forget to provide a suitable title to the table or else the table won't be submitted. Then you can click on Submit Table Form.

4.2 Bar Graph

The screenshot shows a web browser window with the URL 10.100.20.11/reports/junk/41/index.php. The page title is "Micello Monthly Report". The main heading is "Micello's Monthly Report Generator". There are "Reset" and "Sign Out" buttons. Below the heading, it says "Create the following filters:". There are three tabs: "Table", "Bar Graph", and "Maps". The "Bar Graph" tab is selected. The "Bar Graph Form" is displayed with the following fields:

- X-parameter (segregating criteria): Country
- Y-axis parameter (type): All
- Any other filter over the whole bar graph: None
- Parameter Value (if any):
- Title:

There is a "Submit Bar Graph Form" button. Below the form, it says "If finished adding various features, proceed to report generation:". There is a "Generate Report" button. The time and date are 11:28:33 AM, July 16, 2012.

The bar graph form generates a bar graph according to user specified criterias. The first drop down option, X-parameter asks for the segregating criteria using which various maps are to be divided. Y-axis parameter asks for the type of maps to be included which, by default, is all. Suppose you want to include only Shopping Malls then Shopping Malls can be selected. Lastly, the third and fourth options asks for any other filters over the whole bar graphs. *Eg.*, only the maps of USA are to be included - then an overall filter of Country with Parameter Value as United States can be given. Provide the Bar Graph a title and proceed to Submit Bar Graph Form.

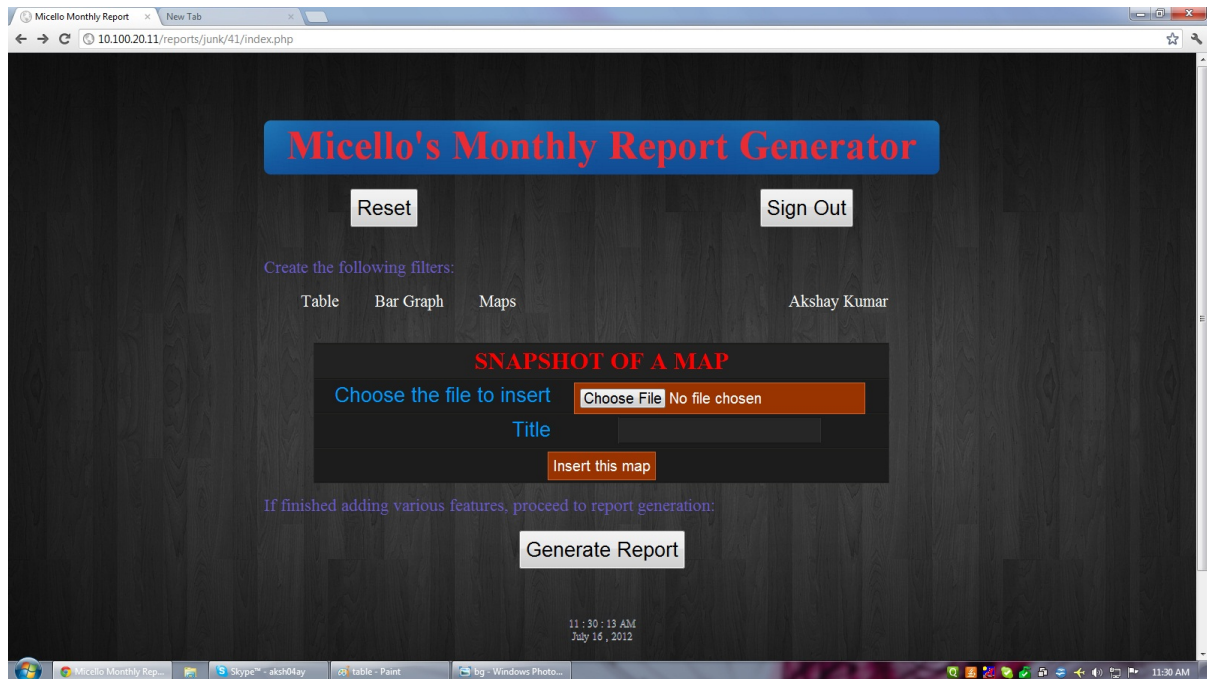
Suppose for example, you want a bar graph depicting the category-wise distribution of shopping malls of USA. Then X-parameter should be Category. Y-axis parameter should be Shopping Malls. The third and fourth parameter should be Country and United States. Lastly, provide the bar graph a relevant title like Category wise distribution of Shopping Malls in USA:-

The close-up shows the "BAR GRAPH FORM" with the following values filled in:

- X-parameter (segregating criteria): Category
- Y-axis parameter (type): Shopping Mall
- Any other filter over the whole bar graph: Country
- Parameter Value (if any): United States
- Title: Category Wise distribution of Shopping Malls

There is a "Submit Bar Graph Form" button.

4.3 Maps



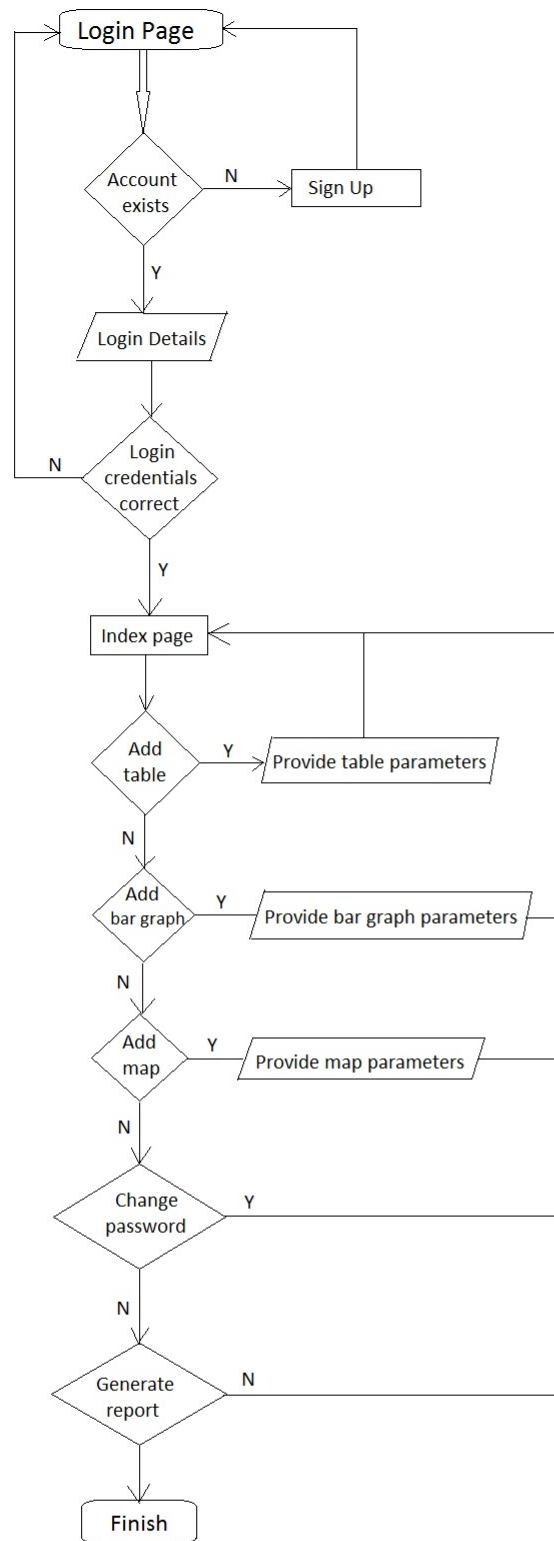
Simply, choose the snapshot of the map which is to be uploaded and upload it providing a relevant title.

4.4 Login feature of the frontend

The url provides a login based access as shown in the snapshot. Sign up and then login. You can change your password after signing in once.



5 Backend of the report generator



The backend starts with the login form which provides access to the main page. At the backend, the user name and other credentials are stored in form of a session variable. Username and corresponding password are verified from a table user and if matched, the access is provided to the main page. The encryption standard used for encrypting the password is MD5.

Both for login page and the main (index) page, jquery has been used to display only relevant forms.

A separate table meta contains the metadata about the various tables, bar graphs, etc. which are to be imported into the report. The schema of meta table is as follows:

Field	Type	Null	Key	Default	Extra
sno	int(11)	NO	PRI	NULL	auto_increment
name	varchar(20)	YES		NULL	
type	varchar(20)	YES		NULL	
title	varchar(255)	YES		NULL	

As clear from the schema, meta table stores the type which is to be embedded into the report (table, bar graph or map) and the title of the table. Name is another metadata which is the name of the temporary table created in the database which contains relevant data.

The very first thing to do upon entering the main page is to press reset button. It flushes away the previous stored jargon values. For this, it iterates through the meta table, searches for all the tables whose names are stored in meta table, deletes them and finally clears the contents of meta table itself.

The Submit table button directs you to a php script which creates a table & inserts it into the database. The sql query done is something like:

```
select * from try where (cname like '%$cname%' or ('$cname'='' and cname is null))
```

for all the columns. Then it drops off the unchecked columns. Finally, the table meta is updated with relevant title and type *viz.* table. You are then redirected to the main page.

The Submit bar graph button directs you to a php script which creates a table based on relevant filters & inserts it into the database. Note that the bar graph is generated based on bar graph table when you finally press Generate Report Button. Finally, the table meta is updated with its title and type *viz.* bar graph.

The Submit map button directs you to a php script which uploads the map chosen. The table meta is then updated with its title and type *viz.* map. It also maintains a table with only one entry which is the name of the image.

On clicking Generate report, a tex file is created. It can be used to generate a pdf file or a ps file or a dvi file whichever required.

6 Future Work

- The User Interface of the Report Generator needs to be improved greatly. It should better be provided a more formal look.
- The only filters used now are Table, Bar Graph and Maps. They can be extended to include more features as and when required.
- The form validation feature of various html forms (tables, bar graphs & maps) are poorly developed. More work needs to be done over the javascript of form validation.
- PHP session variables and redirects are also poorly managed. It can be improved a lot.