Chapter: 1

* 1. **Project Profile**

|  |  |
| --- | --- |
| **Project Title:-** | Issue Tracking System. |
| **Project Definition:-** | The Issue Tracking system allows you to log and track issues(Error).  Users visiting your site can log issues, as can the people involved in the development of your project. |
| **Category** | Web Application |
| **Developed For:-** | Dsyn Hub pvt.ltd. |
|  |  |
| **External Project Guide:-** | MR.Brijesh Kaneriya |
| **Team Size:-** | 2 |
| **Team Member Name:-** | 1)Priyanka Shah(181)  2)Alpa Patel(10) |
| **Hardware requirement:-** | 80 GB HDD  1GB SDRAM  Internet Connection(Network Configuration) |
| **Software requirement:-** | Programming Language-Java  Development Environment- Eclipse IDE |
| **Technology:-** | J2EE |
| **Front End:-** | JSP / Servlet [JAVA SERVER PAGE] |
| **Back End:-** | Oracle 10g |
| **Project Duration:-** | 6 months |

Chapter: 2

**2.1 Company Overview**

Dsyn Hub is a design consulting execution housing design businesses and no matters of graphic design, fashion design, media retail and visual merchandising. Dsyn Hub Competency partner at organizational level forPortals and Collaboration, Business Intelligence, Software Development. At Dsyn Hub our focus is to deliver resultdriven solutions for customer business and helping them realize true value of their investments and to ensure we make technology work for them by ensuring great business outcome. Our value add services include infrastructure planning, consulting, architecturalplanning, consulting, sizing and securing organizational information including businesscontinuity (high availability) .

**2.2 Core Values**

**2.2.1 Customers and Customization**

Every solution provided by Dsyn Hub is defined by the customers' Views. Ourendeavor is to provide our clients with a solution that exceeds their expectations.

**2.2.2 Team work**

We provide equal opportunities for all our team members and ensure harmony betweenorganizational growth and an employee's individual growth. Creativity is appreciatedand communication is encouraged. Quite naturally, the results show in the solutions we provide.

**2.2.3 Integrity and transparency**

We are an organization that swears by ethics, honesty and openness in our

functioning.

**2.2.4 Innovation**

At the heart of every technology is human intelligence. Hence, we understand that there is always a different way of doing things. Even when it means that what we intend to do hasn't to do hasn't been done before.

Chapter: 3

**3.1 Introduction to Development Tools and Technology**

**3.1.1 Tools and Technology Used**

3.1.1.1Technology: J2EE

3.1.1.2 Front-End: JSP-Servlet

3.1.1.3 Back-End: Oracle10g

**3.1.2 Hardware Requirements**

Intel Pentium inside Processor

80 GB Hard Disk

1GB RAM.

**3.1.3 Tools Used**

Eclipse Helios

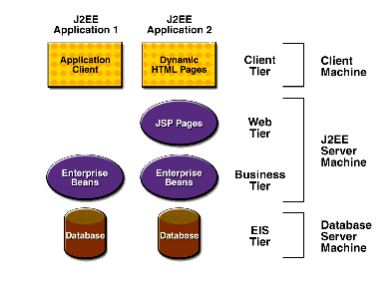
Oracle Query Browser

Microsoft Visio 2007

Apache Tomcat 7.00

# 3.1.1.1[Introduction to J2EE](http://javahowtobyiroshan.blogspot.com/2011/07/introduction-to-j2ee.html)

J2EE or Java 2 Platform, Enterprise Edition is one of the most powerful ways of developing Multitiered web applications. The J2EE platform uses a distributed multitiered application model for enterprise applications. Application logic is divided into components according to function, and the various application components that make up a J2EE application are installed on different machines depending on the tier in the multitiered J2EE environment to which the application component belongs. multitiered application can be describes as in the figure below.



As this is about web application development, there is always two sides as Client & Server. In the client side the java pages, html pages are running & in the server side, Java Server Pages (JSP) , Servlets, Enterprise Java Beans (EJB) are running. So tiers made them separate from each other & also give a clear way to communicate each other.

**3.1.1.2 Java Server Pages – Introduction**

*JSP (Java Server Pages)* is a standard for developing interactive Web applications (pages containing dynamic content). A JSP web page (recognizable by the *.jsp* extension) may display different content based on certain parameters (information stored in a database, the user preferences, ..), while a classic webpage (with the .htm or .html extension) will continuously display the same information.

* JSP is actually a powerful scripting language (interpreted language) *executed on the server side* (like CGI, PHP, ASP, ...) and not on the client side (unlike scripts written in JavaScript or Java applets which run in the browser of the user connected to a site).
* JSPs are integrated in a web page in HTML using special tags which will notify the Web server that the code included within these tags are to be interpreted. The result (HTML codes) will be returned to the client browser .
* Java Server Pages are part of a **3-tier architecture**: where a server supporting the Java Server Pages (generally referred to as **application server**) will act as a mediatior between the client browser and a database (generally referred to as **data server**). JSP provides the necessary elements for the connection to the database management system and allow the manipulation of data through SQL.



## 3.1.1.2 Introduction to Java Servlets

### Objective

Servlet technology was the original Java based solution for web development however due to the problems of maintaining the HTML within the Java code they were never a great success. JSP was the solution to this. However as we will see later there is still a place for Servlets in Java Server development.

This topic looks at the architecture of a Servlet and how to go about writing one.

### What is a Servlet ?

### A Servlet is quite simply a java class that adheres to the general model of a Servlet as defined by the Servlet API. A Servlet Container, also known as a Servlet Engine translates requests from whatever protocol is being used into objects that the Servlet understands, and also gives the Servlet an object which it can use to send a response. This container is also responsible for managing the lifecycle of a Servlet.

Now we have already met the idea of a JSP Container which manages JSP execution. In fact the JSP Container is only responsible for the rewriting of a JSP page to a Servlet and then allows the Servlet Container handle the actual execution.

[**3.1.1.3]Oracle Database 10g Express(As a Back-end-Tool)**

**What is Oracle Database 10g Express Edition?**

* Oracle Database 10g Express Edition (Oracle Database XE) is a free, downloadable version of the world's most capable relational database.
* Oracle Database XE is easy to install and easy to manage.
* With Oracle Database XE, you use the Database Home Page, an intuitive browserbased interface, to administer the database; create tables, views, and other schema objects; import, export, and view table data; run queries and SQL scripts; and generate reports.
* Oracle Database XE includes Oracle HTML DB 2.1, a declarative, graphical development environment for creating database-centric Web applications. In addition to HTML DB 2.1, you can use popular Oracle and third-party languages and tools to develop your Oracle Database XE applications.

**3.2 Introduction To project**

**3.2.1 Definition**

An issue tracking system (ITS) is a software application that allows an enterprise to record and follow the progress of every problem or ”issue” that

a computer system user identifies until the problem is resolved. With an ITS,

an “issue”, which can be anything from a simple customer question to a detailed technical report of an error or bug, can be tracked by priority status, owner, or some other customized criteria.

An ITS generally provides the user with a way to report an issue, track progression towards its resolution, and know who is responsible for resolving the issue. It also allows the manager of the system to customize the tracking procedure so that unnecessary documentation on the part of the problem solvers does not become a waste of time. Many kinds of enterprises use ITS applications, including software developers, manufacturers, IT help desks, and other service providers.

**3.2.2 Objectives**

The main objectives of the proposed system are to help the Client. This System Solve Problems (Issue) of coming Issue during Execution of Project. The system can be handy to the user in the following reasons:

* To provide quick and efficient means for performing Project related activities and to effortlessly generate report of the system.
* To generate the different types of reports.
* Give Report day by Day.

The “Issue Tracking System” of the organization is developed to overcome the most of the problems occurring in the manual system by computerizing the existing system. The features of the newly proposed computerized system are described in brief as below:

After computerizing the system, the owner of the organization or the user of the system can finish their work in least amount of time and efforts. The computerized system has many gains and efforts, which the manual system can’t give, in any type of situations.

Computerized systems are most helpful in dealing with areas where database comes into the existence. A computer can hold large amount of data in its storage devices and it can operate at very high speed. The user can put the entire information in the computer and can be able to perform any type of task which when done manually is tedious and time consuming

Some of the features of the proposed system are given below:

1. Removal of Data Redudancy
2. Data Consistency

**Removal of Data Redundancy:**

Due to the interrelation between the tables and proper assignation of primary keys and foreign keys a data except primary key will never be repeated and hence will cause extremely low redundancy compared to the existing system. This will cause easy data recording and low time consuming.

 **Data Consistency:**

Due to Proper interrelationship and required validations on each and every field any invalid data will not be entered in the system and also due to very less data redundancy any one change in the data will be reflected in every place of the System.

**3.3Business Process Description**

* + My Application Provide Issue Tracking System using jsp servlet.
  + Administrator is authorize to handle all the task of manage all the information on Application.
  + This System Provide Issue Tracking Service such as comming in execution of Any Project.
  + So, this system has two types of users…
* Admin
* User

**Admin**

* + Admin can login in the System and manage all the details such as Manage Project, Manage Module, Manage Task, Manage Issue, Manage Assign Project, and Manage Assign Issue and manage all Users.
  + Administrator can change own password in the system, View All Details of User.
  + **Tasks of Administrator…**
    - He can Login into System.
    - He can insert Update and Delete User Information.
    - He can manage crud Operation of Project, Module, Task as well as Issue.
    - Admin can Assign Role to Employee.
    - Admin can Assign Project to Project Leader as well as Assign Issue to Developer.
    - Admin can Change Password.
    - Admin can generate Report.

**User**

Issue (Error) is handling by User so Every User has Specific User Role such as Developer, Project Leader, and Tester.

**Developer**

* **Task Of Developer**
* Developer can Login in System.
* Developers can Change Password as well as he can give feedback to Admin.
* He can only solve Issue .
* Developer can Change his Profile as well as view his Profile.

**Project Leader**

* **Task Of Project Leader**
* He can Login into System.
* He can Change his Password as well as he can he can give Feedback to Admin.
* He can add project,module,task as well as issue.
* He can Assign issue to Developer.
* He can add issue, assign issue, and solve issue.
* He can view his profile as well as change his profile.

**Tester**

* **Task of Tester**
* He can Login into System as well as Change his Password.
* He can give feedback to Admin.
* He can only add issue and find issue and either closes the bug or reopen it with additional information.
* He can view his Profile as well as update Profile.

Chapter: 4

**4.1 Existing System**

The organization in which all the works are done manually the management approved the need of computerization of system.

**STUDY OF CURRENT SYSTEM**

* Existing system is based on manual work and all the process are done manually, so they maintain registers and files for recording all the details of the Issue.
* Every User’s information was stored in Excel sheet.
* No source for viewing project progress for client.
* Submitting day to day report to seniors by mailing manually.
* There is no security.
* We can’t know how many Employee work on issue as well as Total client.
* We can’t assign Employee Role Properly.
* We can’t categories project such as which issue belong to which Module or Project.
* Documents were submitted using mails.
* Existing system is manual. So it increases the chances of errors.
* Search for any information is done manually, so it is time consuming and tedious.
* It is Manual so we can’t know Project’s or issue’s Status.

**4.2 Need For New System**

* We have work of developed system. The system has been developed to be very user friendly be introducing help feature guiding the user about what to do next at each step.
* The new system will save the time of the Employee and increase efficiency of the Employee.
* Such computerized system that human side of the work very systematically, provided information in a very small period of time and help very much in decision.

**Requirements of New System**

1. Provide the solved issue at right time.
2. Provide report day to day to client.
3. Reduced person dependency.
4. Effective reporting and presentation of required data.
5. Security.
6. Assign Role to Proper Employee.
7. Easy verification/validation process.
8. Easy and Fast Searching of User and Project, Module, task, Issue.
9. Know Status of an issue.
10. **Prioritization.**
11. **Improved control of a project.**

**4.3 System Analysis**

System analysis is the way of studying a system with an eye on solving its problem using computer. It is the most essential part of the development of a project of a system analysis. System analysis consists of system element, process and technology.

To analyze a system, has to study the systems in details. The analyst has to understand the functioning and concept of the system in detail, before design the appropriate computer based system that will meet all the requirements of the existing system. The system analyst has to carry out a customary approach to use the computer for problem solving.

System analysis includes the following basic concepts

* Preliminary investigation
* Requirements specification
* Feasibility study
* Detailed investigation
* Drawing up of strategies
* Design and coding
* Testing and training
* Implementation

The above steps constitute the logical framework for the system analysis. After the preliminary investigation and feasibility study, the scope of the defined and comparable items are set forth and hence detailed investigation is executed. This allows the system analyst to comprehend the full scope of the project. Soon after the implementation of the newly developed system, followed by the training of the users, the system analysis is included.

4.3.1 Preliminary Investigation:

A request to receive assistance from information system can be made for many reasons, but in case a manager, employee or system specialist initiates the request. When that request is made, the first system activity preliminary investigation begins. The activity has three parts

Request clarification: the request from employee may not be well stated. Sometimes the request may not be well defined. Therefore before any system investigation can be considered, the project request must be examined to determine precisely the actual requirements of the organization.

* Feasibility study: the basic idea of feasibility study is to determine whether the requested project is feasible.
* Request approval: all projects that are requested are not desirable or feasible .some organization receive so many projects requests from employee that only a few of them can be pursued. However those projects that are feasible and desirable should put into a schedule. The management decides request that are most important. After a project request is approved the cost priority, the completion time and the personal required are estimated. Once the request is approved, the collection of data and determination of requirements can be started.

4.3.2 Requirement Specification

The primary goal of the system analyst is to improve the efficiency of the existing system. For that the study of specification of the requirements is very essential. For the development of the new system, a preliminary survey of the existing system will be conducted. Investigation done whether the up gradation of the system into an application program could solve the problems and eradicate the inefficiency of the existing system.

**4.3.3 Feasibility Study**

The initial investigation points to the question whether the project is feasible. A feasible is conducted to identify the best system that meets the all requirements. This includes an identification description, an evaluation of the proposed systems and selection of the best system for the job.

The requirements of the system are specified with a set of constraints such as system objectives and the description of the outputs. It is then duty of the analyst to evaluate the feasibility of the proposed system to generate the above results. Three key factors are to be considered during the feasibility study.

**4.3.4 Operation Feasibility**

An estimate should be made to determine how much effort and care will go into the developing of the system including the training to be given to the user. Usually, people are reluctant to changes that come in their progression. The computer initialization will certainly affected the turn over, transfer and employee job status. Hence an additional effort is to be made to train and educate the users on the new way of the system.

**4.3.5 Technical Feasibility**

The main consideration is to be given to the study of available resources of the organization where the software is to be implemented. Here the system analyst evaluates the technical merits of the system giving emphasis on the performance, reliability, maintainability. By taking the consideration before developing the proposed system, the resources availability of the organization was studied. The organization was immense computer facilities equipped with sophisticated machines and the software hence this technically feasible.

**4.3.6 Economic Feasibility**

Economic feasibility is the most important and frequently used method for evaluating the effectiveness of the proposed system. It is very essential because the main goal of the proposed system is to have economically better result along with increased efficiency. Cost benefit analysis is usually performed for this purpose. It is the comparative study of the cost verses the benefit and savings that are expected from the proposed system. Since the organization is well equipped with the required hard ware, the project was found to be economically.

**4.3.7 Data GatheringRequirement Gathering**

For gathering basic needs and functionalities, I was meeting to the programmer of it codes personally to understand required functionalities, also noted down some useful features to be included. By analyzing the information we understand following process requirement.

**Understand different type of user who uses this System**

How many users are going to use this system?

How many types of user going to use this system?

What is the knowledge level of the user?

Provide authorization and separate their functionality?

**Understanding the process of Software**

What are the forms to be included?

How to manage all Forms?

How to manage Users creation Process?

How to manage Registration Process?

How to manage Result process?

How to manage login and logout process?

How to manage all forms linked with each other?

Chapter: 5

**SYSTEM DESIGN**

**5.1 Users**

* Administrator
* Project Manager
* Developer
* Tester
  1. **Modules**
* Login
* Registration
* Project Management
* User Management
* Issue Management
* Task Management
* Module Management
* Status
* Priority

**Data Structures:**

* System is considered perfect if it generate perfect reports.
* System provides facility for sorting and searching.
* Fast retrieval of data.
* Primary key, foreign key and not null constraints have been given in the database.

**PROCESS LOGIC OF MODULE:**

**1.LOGIN:**

**T**he users like Tester, Administrator ,Developer and Project Manager will enter into our project with their user name and password. These different users will be given rights according to their role in their project. When login, all users will be able to have the details such as:

* Issue id
* Issue name
* Project that the issue belonged to
* Status of the issue
* The person who assigned the issue
* The person for whom the issue is assigned
* Date of the issue assigned
* Attachment for the reference of the issue

Different tasks will be available to user according to his/her role in project, tasks likes adding issue, changing issue status. The user can able to delete or can mark it as ‘checked issue’ by selecting the appropriate issue. All users will be having searching criteria in their home page.

The searching criteria will be done according to:

* Project
* issue assigned by the person
* issue assigned to the person
* Status of the issue
* Priority of the issue

**2.PROJECT MANAGEMENT:**

Project details are maintained by the Administrator.

Some details are also maintained by ProjectManager.

The project details are:

* Project name
* Project description
* Developers assigned to the project
* Start date of the project
* Estimate end date of the project

The developers can view the details of the project they need to work on. None can view the details of the project that they not involved in. we are providing high security options.

**3.USERMANAGEMENT:**

All the user details are viewed only by the administrator

The details of the user are:

* Name of the user
* Email id of the user
* Category of the user such as tester, administrator and developer, project manager
* Username of the user, the emailed is used for the login purpose of the user. It should be unique
* Password.

Projects assigned to the User. An User can involve in any modules of different projects. Thus we have to keep the details of the projects assigned to that particular User.

**4.ISSUE MANAGEMENT:**

Issue maintenance is done by the module add issue. Administrator,Project manager and tester will have the authority to add a issue.

* The issue details are:
* issue name
* issue description
* Name of the project, where the issue belongs to
* Name of the person who assigned the issue
* Name of the developer who assigned for fixing the issue
* Priority of the issue
* Status of the issue
* Assigned date for the issue
* Date till the issue need to be closed
* Attachments, we can include a file to refer the issue. It may be of type word, image, or snapshot. It will give the clear view about the issue.

**5.TASK MANAGEMENT:**

Task management is done by administrator. No other users are allowed to be in this module. It consists of all details about the employee details, project details, priorities and statuses. Administrator can add or edit or delete the details of the employee, project, priorities and statuses. This module will also manage the rights of different users according to their role in system.

**6.STATUS MODEL:**

It consists of type of status for the issue as well as for project.

The statuses may be open, fixed, closed, in progress, hold. It is assigned by the administrator. Administrator can include any type of new status according to the flexibility of the issue flow.

**7.PRIORITY:**

It consists of type of priority for the issue.

The priority may be high, low, medium, lowest and normal. It is assigned by the administrator or by the user who adds the issue. Administrator can include any type of priorities according to the flexibility of the issue flow.

Entity Relationship Diagram

**5.3 Entity Relationship Diagram**

#### Data models are tools used in analysis to describe the data requirements and assumptions in the system from a top-down perspective. They also set the stage for the design of databases later on in the SDLC.

#### There are three basic elements in ER models:

#### Entities are the "things" about which we seek information.

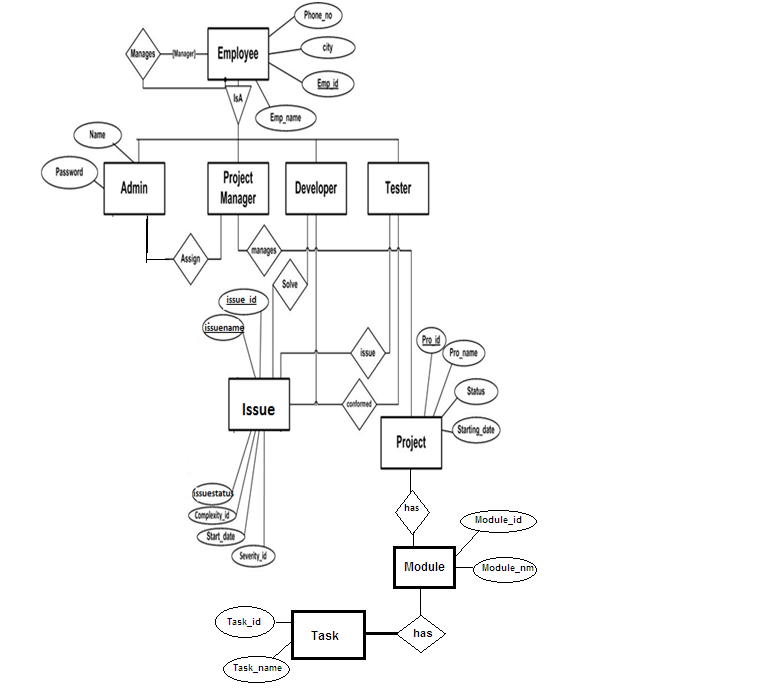
#### Attributes are the data we collect about the entities.

#### Relationships provide the structure needed to draw information from

#### multiple entities.

The basics used here are:

|  |  |
| --- | --- |
|  | Entity Class |
|  | Relationship Type |
|  | Attribute |
|  | Multivalve Attributes |
|  | Key Attribute |
|  | Weak Entity Class |
|  | Derived Attribute |



**[fig.5.3 E-R Diagram]**

Data Flow Diagram

**5.4 Data Flow Diagram**

**Symbol Used in Data Flow Diagram ( DFDs )**

The data flow diagrams are pictorial or graphical representation of the outline of the system study. The data flow diagram covers all the processes and data storage area which takes place during any transaction in the system. The data flow diagrams are functionally divided into context level, Zero level, First level and Second level data flow diagrams.

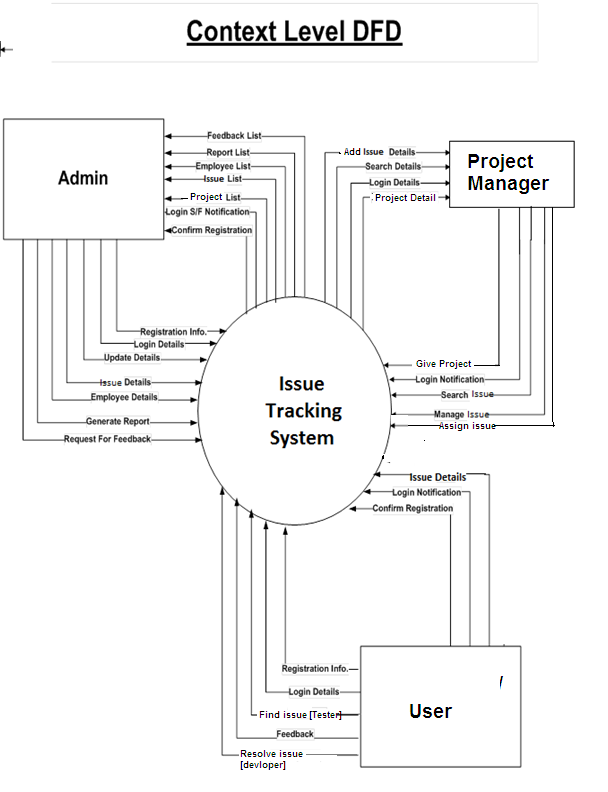
**Symbols used in DFDs:**

**Process:** Here flow of data is transformed. E.g. Purchase of items, update inventory file, etc.

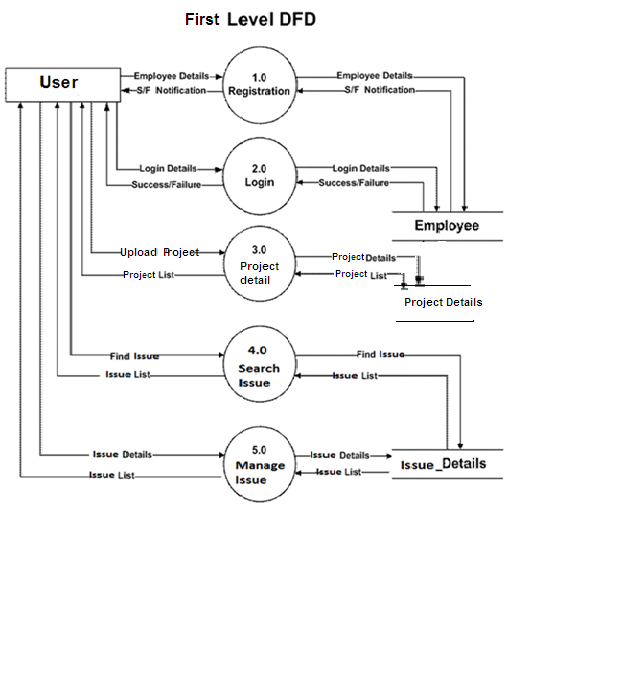
**External Entity:** A source or destination of data which is external to the system. E.g. Customer, Supplier etc.

**A data flow:** It is packet of data. It may be in the form of document, letter etc.

**Data store:** Any store data but with no reference to the physical method of storing.



**[5.4.1 fig Context Level DFD]**



**[5.4.2fig. First leval Data Flow Diagram for User]**

**Second Level**



**Third Level DFD:-**

****

**5.5 Integration Diagram**

**5.5.1 Use Case Diagram**

**Admin Use Case**



**Project Leader Use Case**



**Developer Use Case**



**QA / Tester Use Case**



Chapter: 6

Data Dictionary

**DATA DICTIONARY**

|  |  |
| --- | --- |
| 1.Tbl\_country |  |
| Description: Contain the Country |  |

**Objective:** The Country table stores the Country.It uses of Registration table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field Name** | **Description** | **Type** | **Size** | **Constraint** | **Default Value** |
| country\_id | Uniquely Identifies Country | Number | 3 | PK | Not Null |
| country\_name | Specifies Name of Country | Varchar2 | 50 |  | Not Null |

**Validations** : The table does not allow to the user to enter blank field those are required as those fields are not null and it will show an error message if the any of the field which is required is kept null by the user while doing any transaction using this table.

This table’s Primary key has use of registration table on Ajax.

**Primary key:**country\_id

**Foreign key:** no reference in this table

|  |  |
| --- | --- |
| 2.Tbl\_state |  |
| Description: Contain the State |  |

**Objective:** The Country table stores the Country.It uses of Registration table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field Name** | **Description** | **Type** | **Size** | **Constraint** | **Default Value** |
| state\_id | Uniquely Identifies State | Number | 3 | PK | Not Null |
| state\_name | Specifies Name of State | Varchar2 | 50 |  | Not Null |
| country\_id | Uniquely Identifies Country | Number | 3 | FK | Not Null |

**Validations** : The table does not allow to the user to enter blank field those are required as those fields are not null and it will show an error message if the any of the field which is required is kept null by the user while doing any transaction using this table.

This table’s Primary key has use of registration table on Ajax.

**Primary key:**state\_id

**Foreign key:** reference on country\_id in country table

|  |  |
| --- | --- |
| 3.Tbl\_city |  |
| Description: Contain the city |  |

**Objective:** The City table stores the City.It uses of Registration table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field Name** | **Description** | **Type** | **Size** | **Constraint** | **Default Value** |
| city\_id | Uniquely Identifies City | Number | 3 | PK | Not Null |
| city\_name | Specifies Name of City | Varchar2 | 50 |  | Not Null |
| state\_id | Uniquely Identifies State | Number | 3 | FK | Not Null |

**Validations** : The table does not allow to the user to enter blank field those are required as those fields are not null and it will show an error message if the any of the field which is required is kept null by the user while doing any transaction using this table.

This table’s Primary key has use of registration table on Ajax.

**Primary key:**city\_id

**Foreign key:** reference on state\_id in state table

|  |  |
| --- | --- |
| 4.tbl\_user\_role |  |
| Description: Enter the user role |  |

**Objective:**The User\_role table stores information about user role .User role:Admin,Project Manager,Team Leader,Developer,Tester.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field Name** | **Description** | **Type** | **Size** | **Constraint** | **Default Value** |
| user\_role\_id | Identifing ROLE of user | Number | 3 | PK | Not Null |
| user\_role\_description | Identifies ROLE of user | Varchar2 | 50 |  | Not Null |

**Validations** : The table does not allow to the user to enter blank field those are required as those fields are not null and it will show an error message if the any of the field which is required is kept null by the user while doing any transaction using this table.

**Primary key:**user\_role\_id

**Foreign key:** reference on state\_id in user registration table

|  |  |
| --- | --- |
| 5.tbl\_user\_type |  |
| Description: Enter the user type |  |

**Objective:**The User\_type table stores information about user role .User type:Admin, User.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field Name** | **Description** | **Type** | **Size** | **Constraint** | **Default Value** |
| user\_type\_id | Uniquely Identify Type | Number | 2 | PK | Not Null |
| user\_type | Specifies Type of User | Varchar2 | 50 |  | Not Null |

**Validations** : The table does not allow to the user to enter blank field those are required as those fields are not null and it will show an error message if the any of the field which is required is kept null by the user while doing any transaction using this table.

**Primary key:**user\_role\_id

**Foreign key:** reference on state\_id in user registration table

|  |  |
| --- | --- |
| 6.security Question |  |
| Description: Contain the Security Question |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field Name** | **Description** | **Type** | **Size** | **Constraint** | **Default Value** |
| security\_question\_id | Uniquely Identifies Question | Number | 3 | PK | Not Null |
| security\_question | Specifies Question for Security | Varchar2 | 50 |  | Not Null |

**Objective:**The security question table stores Question for sequrity purpose.

**Validations** : The table does not allow to the user to enter blank field those are required as those fields are not null and it will show an error message if the any of the field which is required is kept null by the user while doing any transaction using this table.

**Primary key:**security\_question\_id

|  |
| --- |
| 7.TBL\_PMT\_Priority |
| Description: Project, Module & Task Priority detailed e.g:High,Low,Medium |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field Name** | **Description** | **Type** | **Size** | **Contraint** | **Default Value** |
| pmt\_priority\_id | Pmt Priority ID | Number | 5 | PK | Not Null |
| pmt\_priority | Pmt Priority | Varchar2 | 50 |  | Not Null |

**Objective:**The PMT\_Priority table stores information about Project, Module & Task Priority.Priority:High, Low, Medium.

**Validations** : The table does not allow to the user to enter blank field those are required as those fields are not null and it will show an error message if the any of the field which is required is kept null by the user while doing any transaction using this table.

**Primary key:**PMT\_Priority\_id

|  |
| --- |
| 8.tbl\_PMT\_Status |
| Description: Contain Project, Module, Task Status for Exa: On Hold, Running, Rejected. |

**Objective:**The PMT\_Status table stores information about Project, Module & Task Status.Status:On Hold, Running, Rejected.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field Name** | **Description** | **Type** | **Size** | **Contraint** | **Default Value** |
| pmt\_status\_id | Pmt Status ID | Number | 5 | PK | Not Null |
| pmt\_status | Pmt Status | Varchar2 | 50 |  | Not Null |

**Validations** : The table does not allow to the user to enter blank field those are required as those fields are not null and it will show an error message if the any of the field which is required is kept null by the user while doing any transaction using this table.

**Primary key:**PMT\_Status\_id

|  |
| --- |
| 9.TBL\_Issue\_Status |
| Description: Contain all issue status for Exa:on hold,running,rejected |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field Name** | **Description** | **Type** | **Size** | **Contraint** | **Default Value** |
| Issue\_status\_id | Issue Status ID | Number | 5 | PK | Not Null |
| Issue\_status | Issue Status | Varchar2 | 50 |  | Not Null |

**Objective:**The issue\_Status table stores information about Issue Status.Status:On Hold, Running, Rejected.

**Validations** : The table does not allow to the user to enter blank field those are required as those fields are not null and it will show an error message if the any of the field which is required is kept null by the user while doing any transaction using this table.

**Primary key:**issue\_Status\_id

|  |
| --- |
| 10.tbl\_issue\_priority |
| Description:Contain all issue Priority for Exa :High, Low, Medium |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field Name** | **Description** | **Type** | **Size** | **Contraint** | **Default Value** |
| Issue\_priority\_id | Issue Priority ID | Number | 5 | PK | Not Null |
| Issue\_priority | Issue Priority | Varchar2 | 50 |  | Not Null |

**Objective:**The issue\_priority table stores information about Issue. Priority:High, Low, Medium.

**Validations** : The table does not allow to the user to enter blank field those are required as those fields are not null and it will show an error message if the any of the field which is required is kept null by the user while doing any transaction using this table.

**Primary key:**issue\_Priority\_id

|  |
| --- |
| 11.tbl\_issue\_type |
| Description: contain the issue type. For Exa: Number Format Exception e. t. c . |

**Objective:**The issue\_priority table stores information about Issue Priority.Priority:High, Low, Medium.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field Name** | **Description** | **Type** | **Size** | **Contraint** | **Default Value** |
| Issue\_type\_id | Issue Type ID | Number | 5 | PK | Not Null |
| Issue\_type | Issue Type | Varchar2 | 50 |  | Not Null |

**Validations** : The table does not allow to the user to enter blank field those are required as those fields are not null and it will show an error message if the any of the field which is required is kept null by the user while doing any transaction using this table.

**Primary key:**issue\_Type\_id

|  |
| --- |
| 12.tbl\_os |
| Description: Contain all the Operating System |

**Objective:**The OS table stores information about Operating System.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field Name** | **Description** | **Type** | **Size** | **Contraint** | **Default Value** |
| os\_id | OS ID | Number | 5 | PK | Not Null |
| Os | OS | Varchar2 | 50 |  | Not Null |

**Validations** : The table does not allow to the user to enter blank field those are required as those fields are not null and it will show an error message if the any of the field which is required is kept null by the user while doing any transaction using this table.

**Primary key:**os\_id

|  |
| --- |
| 13.tbl\_project\_categary |
| Description: Contain all the Project Technology or Language |

**Objective:**The Project Categarytable stores information about Project Technology or Language.For Exa:.Net,Java,ASP.NET

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field Name** | **Description** | **Type** | **Size** | **Contraint** | **Default Value** |
| project\_category\_id | Project Category ID | Number | 5 | PK | Not Null |
| project\_category | Project Category | Varchar2 | 50 |  | Not Null |

**Validations** : The table does not allow to the user to enter blank field those are required as those fields are not null and it will show an error message if the any of the field which is required is kept null by the user while doing any transaction using this table.

**Primary key:**Project\_categary\_id

|  |
| --- |
| 14.tbl\_Feedback |
| Description: Contain all the feedback detail. |

**Objective:**The Feedback table stores information about Feedback of own Website.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field Name** | **Description** | **Type** | **Size** | **Contraint** | **Default Value** |
| feedback\_id | Feedback ID | Number | 5 | PK | Not Null |
| feedback\_name | Feedback name | Varchar2 | 50 |  | Not Null |
| feedback\_Description | Feedback Description | Varchar2 | 100 |  | Not Null |
| feedback\_date | feedback date | date |  |  | Not Null |
| Email\_id | Emailed | Varchar2 | 100 |  | Unique |

**Validations** : The table does not allow to the user to enter blank field those are required as those fields are not null and it will show an error message if the any of the field which is required is kept null by the user while doing any transaction using this table.

**Primary key:**feedback\_id

|  |
| --- |
| 15.tbl\_Projectmaster |
| Description: Contain all details of different Project |

**Objective:**The Project Master table stores information about Different Project.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field Name** | **Description** | **Type** | **Size** | **Contraint** | **Default Value** |
| proj\_id | Project ID | Number | 5 | PK | Not Null |
| project\_name | Project Name | Varchar2 | 50 |  | Not Null |
| project\_description | Project Descritpion | Varchar2 | 200 |  | Not Null |
| project\_estimate\_start\_date | Project Estimate Start Date | Date |  |  | Not Null |
| project\_actual\_start\_date | Project Actual Start Date | Date |  |  | Null |
| project\_estimate\_end\_date | Project Estimate End Date | Date |  |  | Not Null |
| project\_actual\_end\_date | Project Actual End Date | Date |  |  | Null |
| PMT\_priority\_id | Project Priority ID | Number | 5 | FK | Not Null |
| PMT\_status\_id | Project Status ID | Number | 5 | FK | Not Null |
| project\_category\_id | Project Category ID | Number | 5 | FK | Not Null |
|  |  |  |  |  |  |

**Validations** : The table does not allow to the user to enter blank field those are required as those fields are not null and it will show an error message if the any of the field which is required is kept null by the user while doing any transaction using this table.

**Primary key:**Proj\_id

**Foreign Key:**PMT\_priority\_id, PMT\_status\_id, project\_categary\_id

|  |
| --- |
| 16.tbl\_Module |
| Description: Contain all module Detail |

**Objective:**The Module table stores information about Different Project’s Module.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field Name** | **Description** | **Type** | **Size** | **Contraint** | **Default Value** |
| module\_id | Module ID | Number | 5 | PK | Not Null |
| module\_name | Module Name | Varchar2 | 50 |  | Not Null |
| module\_description | Module Descritpion | Varchar2 | 200 |  | Not Null |
| module\_estimate\_start\_date | Module Estimate Start Date | Date |  |  | Not Null |
| module\_actual\_start\_date | Module Actual Start Date | Date |  |  | Null |
| module\_estimate\_end\_date | Module Estimate End Date | Date |  |  | Not Null |
| module\_actual\_end\_date | Module Actual End Date | Date |  |  | Null |
| PMT\_priority\_id | Module Priority ID | Number | 5 | FK | Not Null |
| PMT\_status\_id | Module Status ID | Number | 5 | FK | Not Null |
| proj\_id | Project ID | Number | 5 | FK | Not Null |

**Validations** : The table does not allow to the user to enter blank field those are required as those fields are not null and it will show an error message if the any of the field which is required is kept null by the user while doing any transaction using this table.

**Primary key:**module\_id

**Foreign Key:**PMT\_priority\_id, PMT\_status\_id, proj\_id

|  |
| --- |
| 17.tbl\_task |
| Description: Contain all the task. |

**Objective:**The Task table stores information about Different Project’s &Module’s task.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field Name** | **Description** | **Type** | **Size** | **Contraint** | **Default Value** |
| task\_id | Task ID | Number | 5 | PK | Not Null |
| task\_name | Task Name | Varchar2 | 50 |  | Not Null |
| task\_description | Task Descritpion | Varchar2 | 200 |  | Not Null |
| task\_estimate\_start\_date | Task Estimate Start Date | Date |  |  | Not Null |
| task\_actual\_start\_date | Task Actual Start Date | Date |  |  | Null |
| task\_estimate\_end\_date | Task Estimate End Date | Date |  |  | Not Null |
| task\_actual\_end\_date | Task Actual End Date | Date |  |  | Null |
| PMT\_priority\_id | Task Priority ID | Number | 5 | FK | Not Null |
| PMT\_status\_id | Task Status ID | Number | 5 | FK | Not Null |
| module\_id | Module ID | Number | 5 | FK | Not Null |

**Validations** : The table does not allow to the user to enter blank field those are required as those fields are not null and it will show an error message if the any of the field which is required is kept null by the user while doing any transaction using this table.

**Primary key:**task\_id

**Foreign Key:**PMT\_priority\_id, PMT\_status\_id, module\_id

|  |
| --- |
| 18.tbl\_Project\_User |
| Description: |

**Objective:**The Project user table search information about Different Project’s & Module’s task.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field Name** | **Description** | **Type** | **Size** | **Contraint** | **Default Value** |
| project\_user\_id | Project User ID | Number | 5 | PK | Not Null |
| proj\_id | Project ID | Number | 5 | FK | Not Null |
| user\_id | User\_id | Number | 5 | FK | Not Null |
| role\_id | Role\_id | Number | 5 | FK | Not Null |

**Validations** : The table does not allow to the user to enter blank field those are required as those fields are not null and it will show an error message if the any of the field which is required is kept null by the user while doing any transaction using this table.

**Primary key:**project\_user\_id

**Foreign Key:**proj\_id, user\_id, role\_id

|  |
| --- |
| 19.tbl\_issue\_detail |
| Description: detail of project’s issues |

**Objective:**The issue detail table all information about Different Project’s Issues.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field Name** | **Description** | **Type** | **Size** | **Contraint** | **Default Value** |
| Issue\_id | Issue ID | Number | 5 | PK | Not Null |
| Issue\_title | Issue Title | Varchar2 | 50 |  | Not Null |
| Issue\_desciption | Issue Description | Varchar2 | 500 |  | Not Null |
| Issue\_type\_id | Issue Type ID | Number | 5 | FK | Not Null |
| Issue\_creation\_date | Issue Creation Date | Date |  |  | Not Null |
| Issue\_taks\_id | Issue Task ID | Number | 5 | FK | Not Null |
| user\_id | Issue Post User ID | Number | 5 | FK | Not Null |
| os\_id | OS | Number | 5 | FK | Not Null |
| Issue\_priority\_id | Issue Priority ID | Number | 5 | FK | Not Null |
| Issue\_status\_id | Issue Status ID | Number | 5 | FK | Not Null |
| Issue\_status\_date | Issue Status Date | Date |  |  | Not Null |

**Validations** : The table does not allow to the user to enter blank field those are required as those fields are not null and it will show an error message if the any of the field which is required is kept null by the user while doing any transaction using this table.

**Primary key:**issue\_id

**Foreign Key:**proj\_id, user\_id, role\_id

|  |
| --- |
| 20.tbl\_issue\_assign |
| Description: Assign Issue to a Devloper. |

**Objective:**The issue assign table all information about Different Project’s Issues to assign on which devloper.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field Name** | **Description** | **Type** | **Size** | **Contraint** | **Default Value** |
| Issue\_assign\_id | Issue Assign ID | Number |  | PK | Not Null |
| Issue\_id | Issue ID | Number | 5 | FK | Not Null |
| Issue\_assign\_to | Issue Assign To (User ID) | Number | 5 | FK | Not Null |
| Issue\_reported\_by | Issue Reported By (User ID) | Number | 5 | FK | Not Null |
| Issue\_assign\_date | Issue Reported Date | Date |  |  | Not Null |

**Validations** : The table does not allow to the user to enter blank field those are required as those fields are not null and it will show an error message if the any of the field which is required is kept null by the user while doing any transaction using this table.

**Primary key:**issue\_assign\_id

**Foreign Key:**issue\_id, issue\_assign\_to, issue\_reported\_by.

|  |
| --- |
| 21.tbl\_issue\_resolution |
| Description: Detail of the Solve Issue. |

**Objective:**The issue assign table all information about Different Project’s Issues to assign on which devloper.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field Name** | **Description** | **Type** | **Size** | **Contraint** | **Default Value** |
| Issue\_resolution\_id | Issue\_resolution\_id | Number | 5 | PK | Not Null |
| Issue\_id | Issue ID | Number | 5 | FK | Not Null |
| Issue\_action\_t0 | Issue Action Type | Varchar2 | 100 |  | Not Null |
| Issue\_action\_description | Issue Action Description | Varchar2 | 500 |  | Not Null |
| Issue\_resolved\_by | Issue Resolved By(User ID) | Number | 5 | FK | Not Null |
| Issue\_resolved\_date | Issue Resolved Date | Date |  |  | Not Null |

**Validations** : The table does not allow to the user to enter blank field those are required as those fields are not null and it will show an error message if the any of the field which is required is kept null by the user while doing any transaction using this table.

**Primary key:**issue\_resolution\_id

**Foreign Key:**issue\_id,issue\_resolved\_by.

|  |
| --- |
| 22.tbl\_user\_registration |
| Description: Stores Information about user |

**Objective:** The User\_Registration table stores information about user. It stores general as well as personal information of user. It also stores information about applicants who has given their resume to the center so that staff can be made available when it is required.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field Name** | **Description** | **Data Type** | **Size** | **Constraint** | **Default Value** |
| user\_id | User ID | Number | 5 | PK | Not Null |
| user\_firstname | User First Name | Varchar2 | 30 |  | Not Null |
| user\_middlename | User Middle Name | Varchar2 | 30 |  | Null |
| user\_lastname | User Last Name | Varchar2 | 30 |  | Not Null |
| user\_res\_address | User Residential Address | Varchar2 | 250 |  | Not Null |
| user\_per\_address | User Permanent Address | Varchar2 | 250 |  | Not Null |
| city\_id | City ID | Number | 3 | FK | Not Null |
| user\_gender | User Gender | Varchar2 | 7 |  | Not Null |
| user\_dob | User Date of Birth | Date | - |  | Not Null |
| user\_phoneno | User Phone No | Number | 15 |  | Not Null |
| user\_mobileno | User Mobile No | Number | 10 |  | Not Null |
| user\_email | User Email ID | Varchar2 | 50 | Unique | Not Null |
| user\_password | User Password | Varchar2 | 50 |  | Not Null |
| user\_type\_id | User Role ID | Number | 3 | FK | Not Null |
| security\_question\_id | Question ID | Number | 3 | FK | Not Null |
| user\_question\_ans | Security Question Ans | Varchar2 | 150 |  | Not Null |
| user\_login\_date | User Last Login Date | Date | - |  | Not Null |
| user\_status | User Status | Number | 1 |  | Not Null |

**Validations** : The table does not allow to the user to enter blank field those are required as those fields are not null and it will show an error message if the any of the field which is required is kept null by the user while doing any transaction using this table.

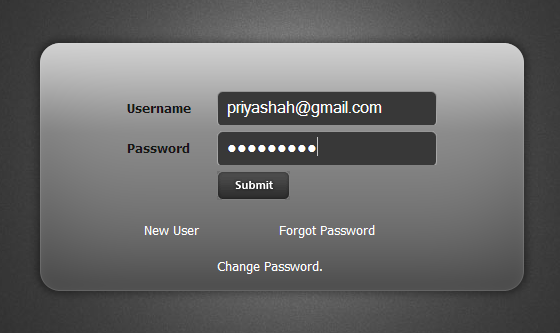
And also Ajax (DWR) validation.

**Primary key:** user\_id

**Foreign key:** city\_id in city table, sequrity\_question\_id insecurity question table

User\_type\_id in user\_type

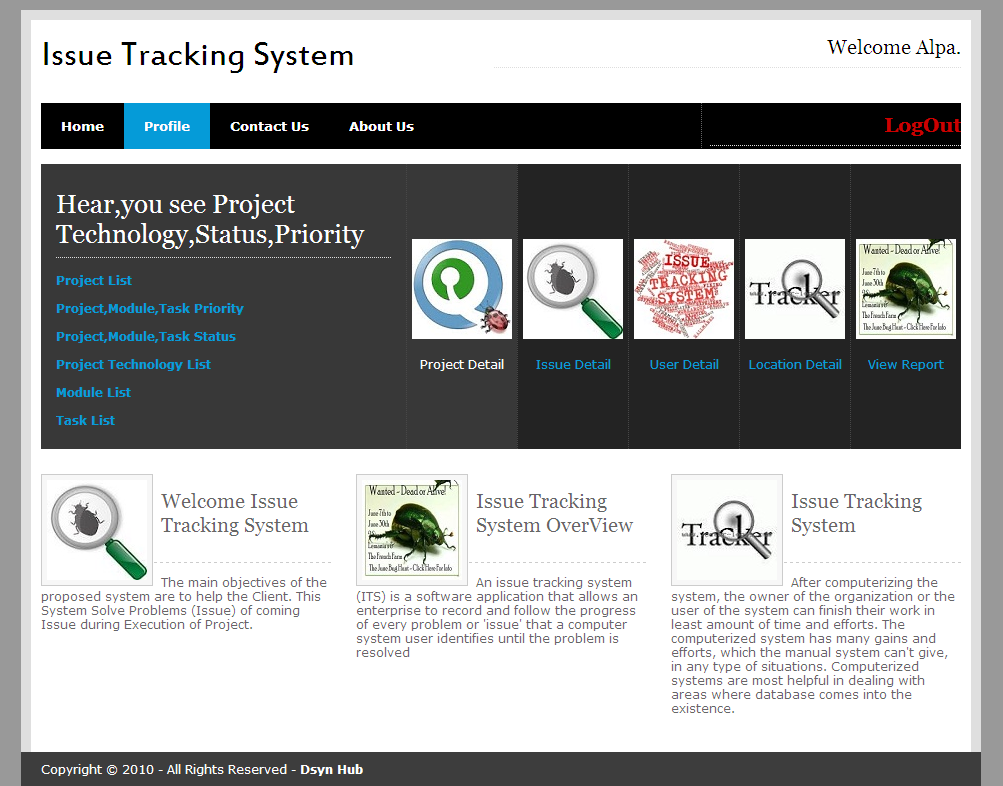
1. **Login Form:-**



The User login screen for logging into the system is shown above. As soon as the user starts the system he/she faces the login screen where the user is asked to enter the userid and password. To log into the system the user have to enter correct username as well as password. After entering correct username and password and by pressing Login the system checks for valid username and password of the user. If the username or password is not valid then it doesn’t allow logging the user in the system and gives a prompting message of invalid username or password.

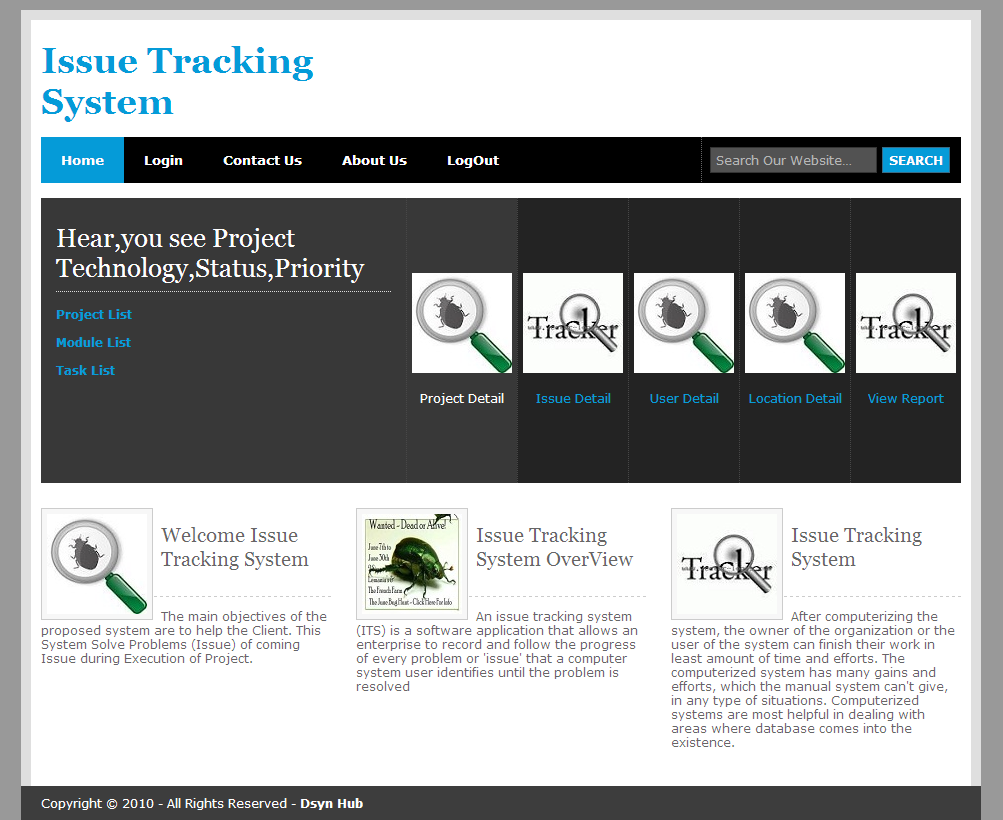
User should be employee of the company.

1. **Admin Home Page**



When admin can be login then this page will be open.

1. **Project Manager Home Page:-**



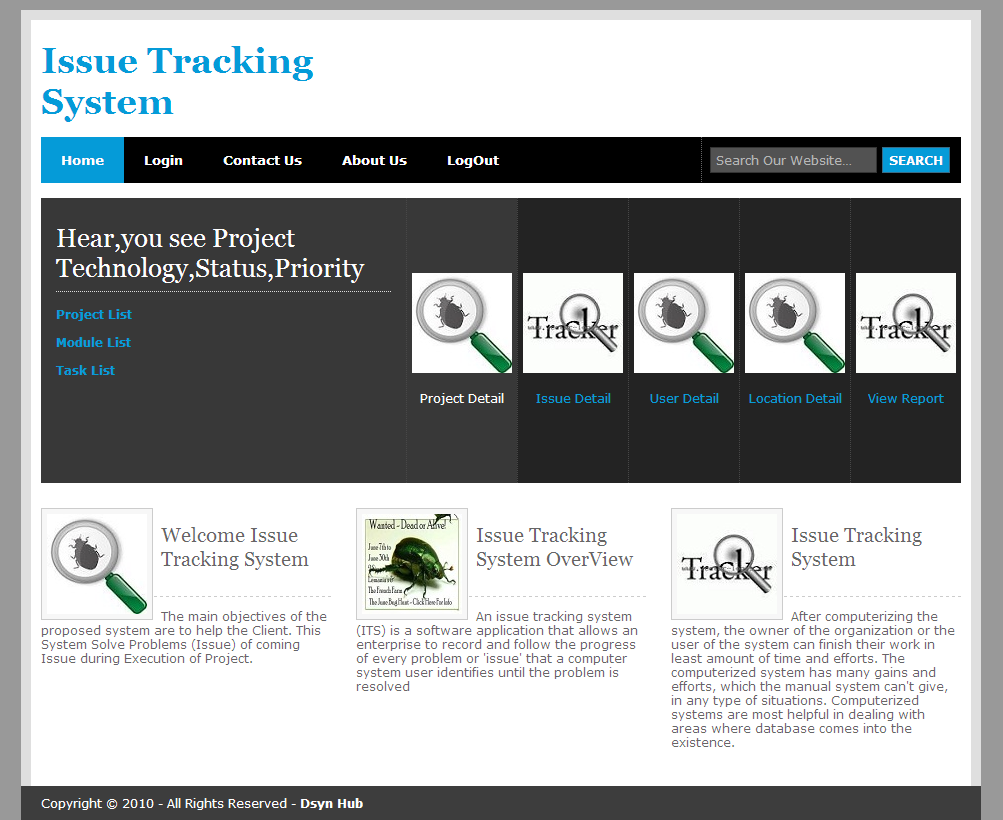
This Home page is Project Manager. This home page handle of the some rights for example:Add the Project , Add the Issue ,e.t.c.

1. **Developer Home Page:-**



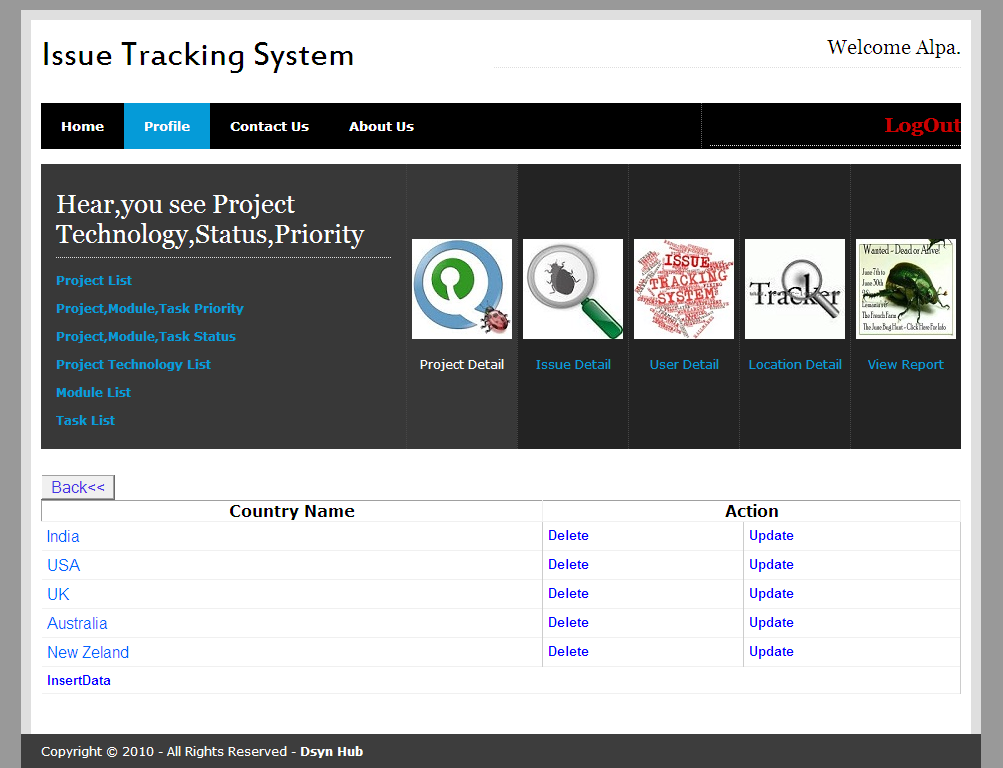
This Home Page is Developer. Devloper can handle the some rights. For exa: View the Project And Issue detail and resolve the Issue.

1. **Tester Home Page:-**



This home page is Tester Side. Tester can handle some rights For Example: Find the issue,and then return back the Project Manager.

1. **Country List Form:-**



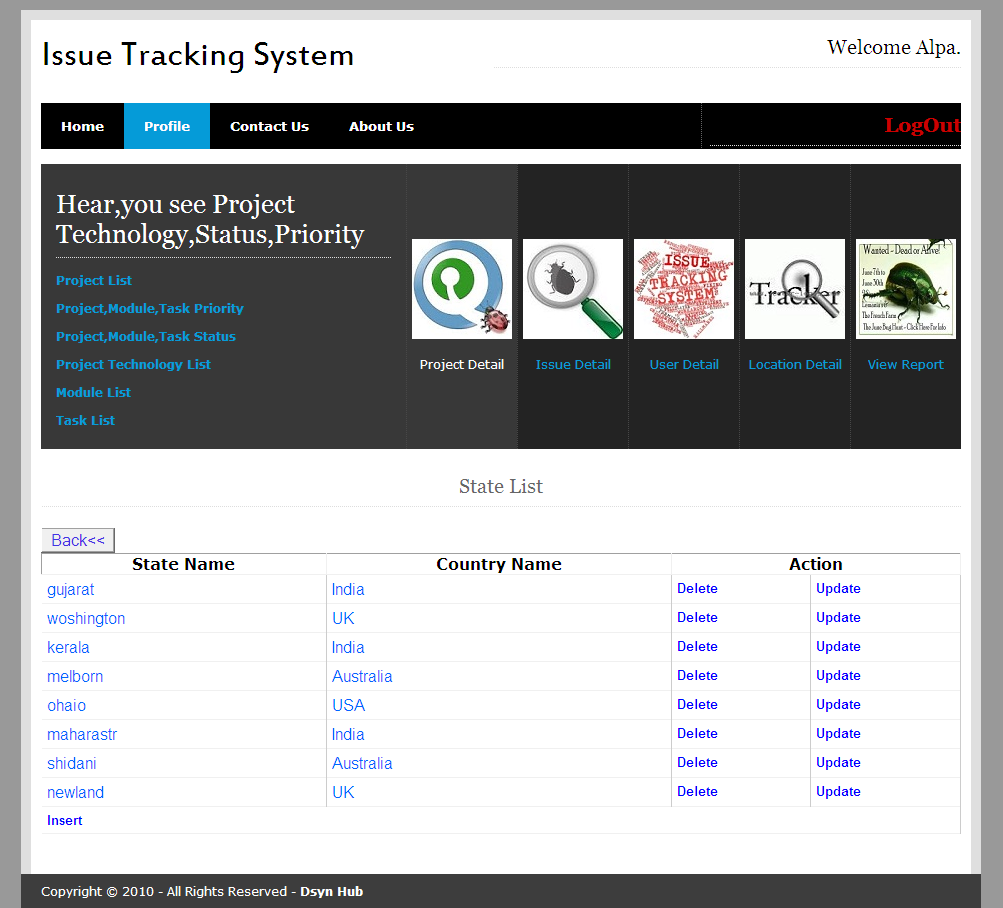
It will give the List of Country name.

1. **Country Insert Form:-**



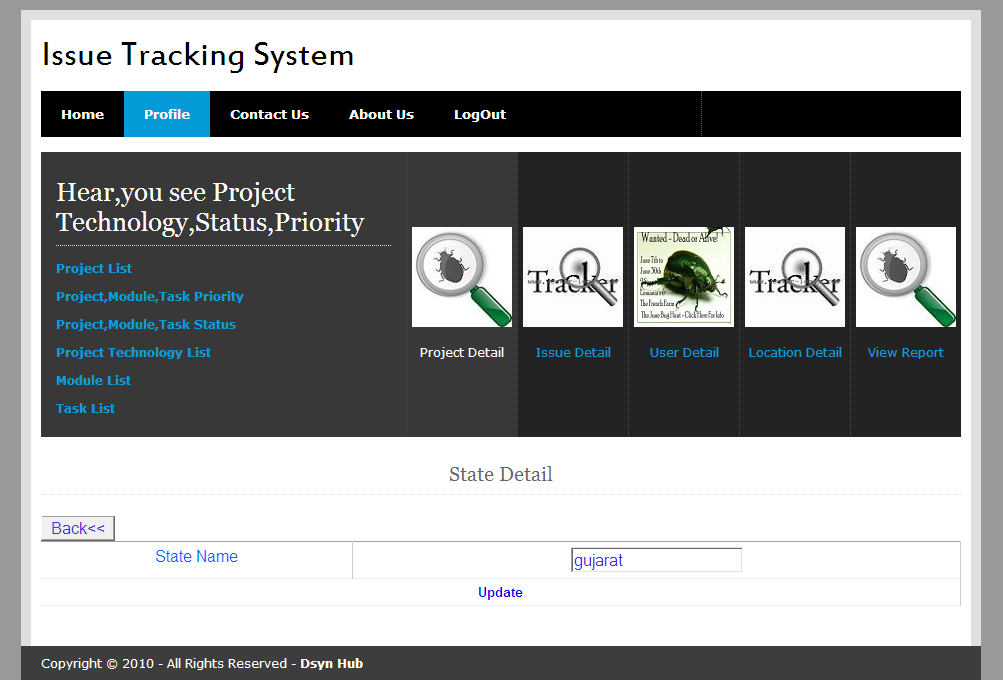
Here you can insert Country.Name.which is useful in registration form.

1. **State List Form:-**



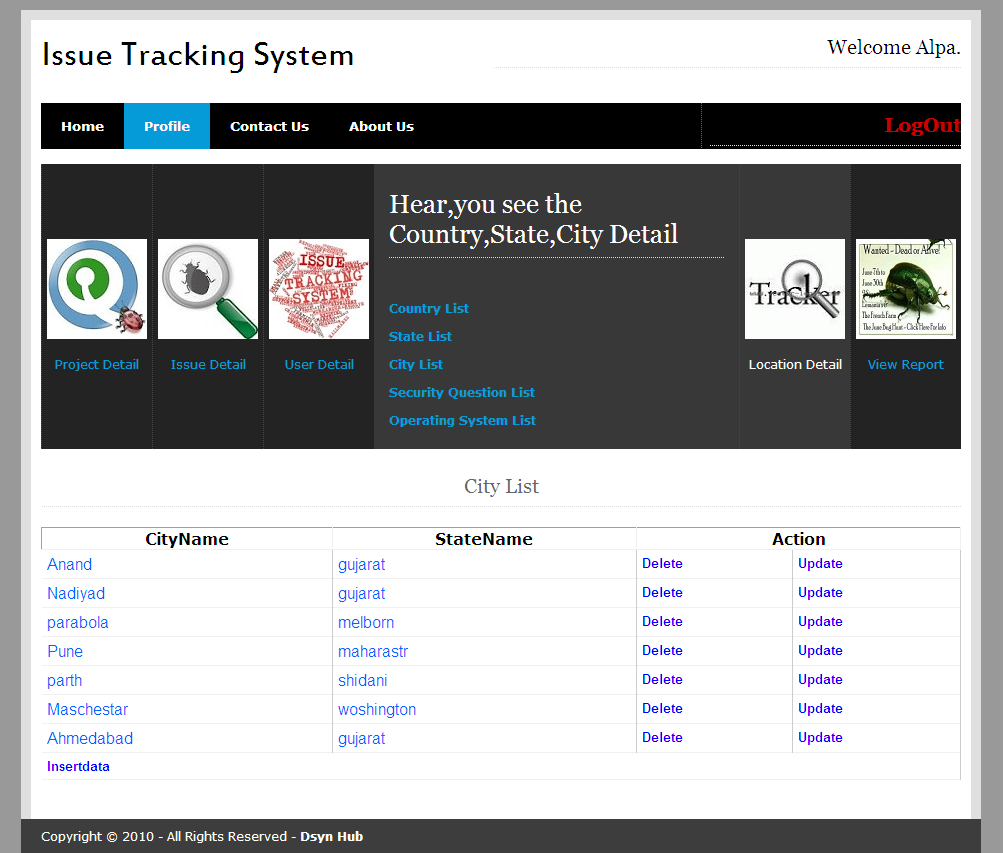
It gives list of State List

1. **State Edit Form:-**



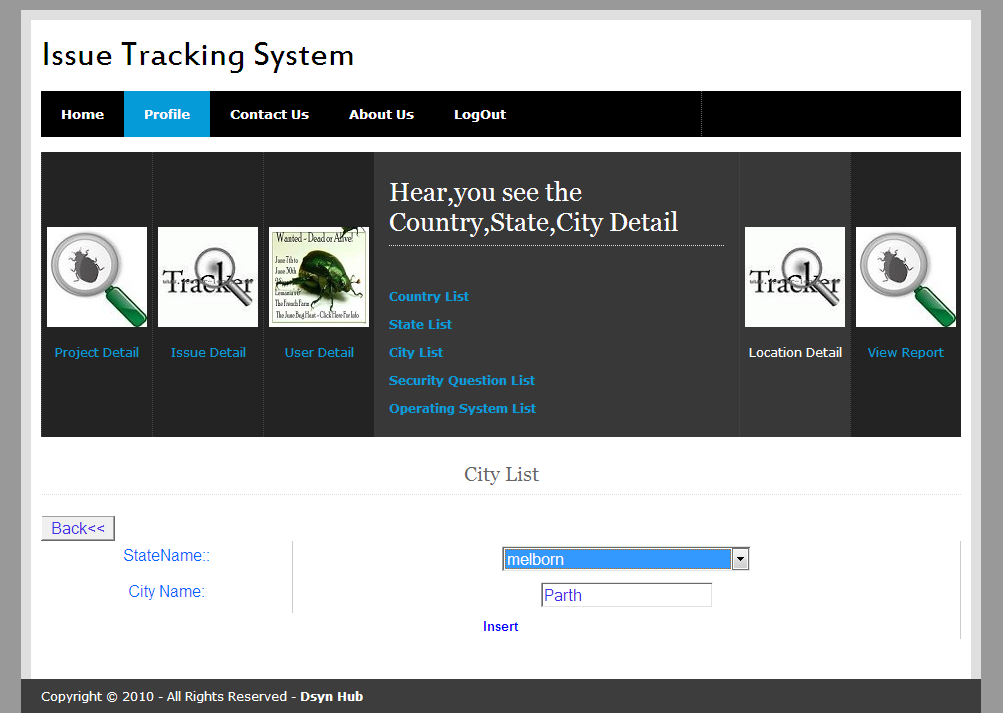
Edit the State name

1. **City List Form:-**



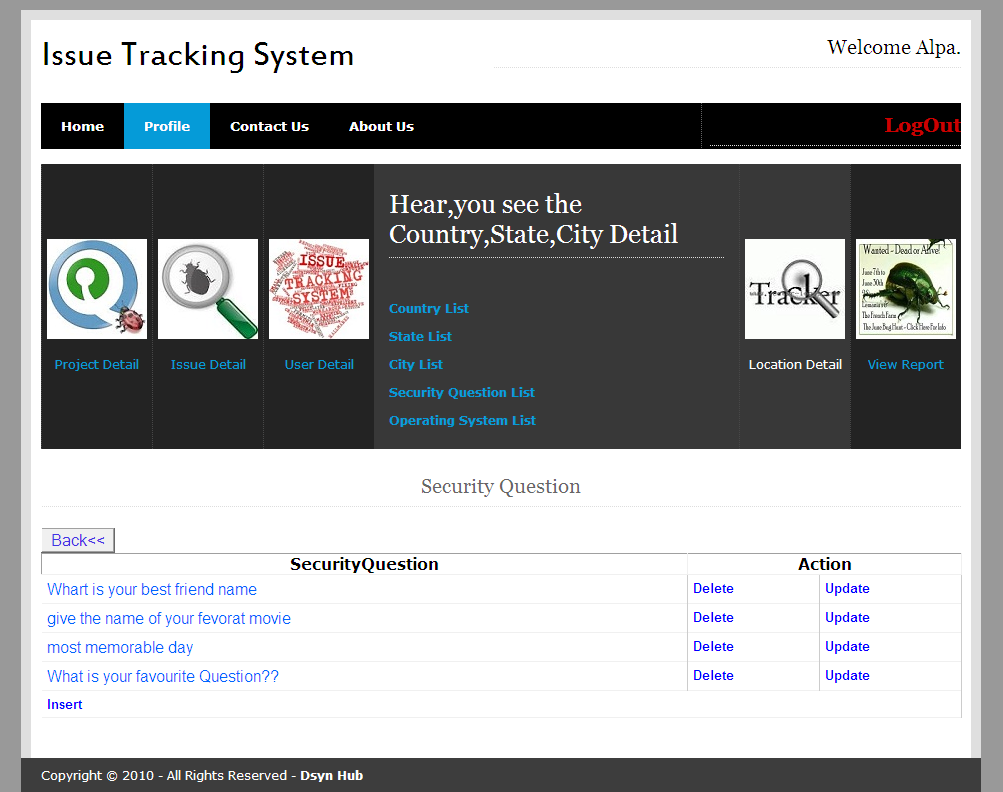
It gives the List of the City name.

1. **City Insert Form:-**

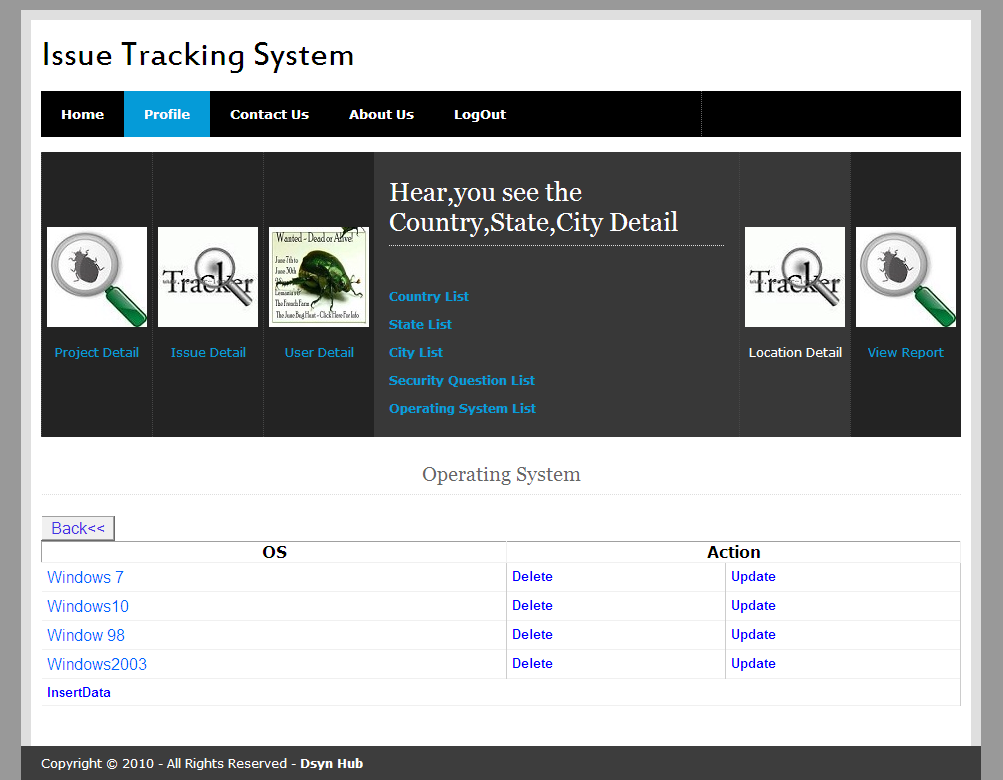


Insert a New City ,only Admin Can do this Process .

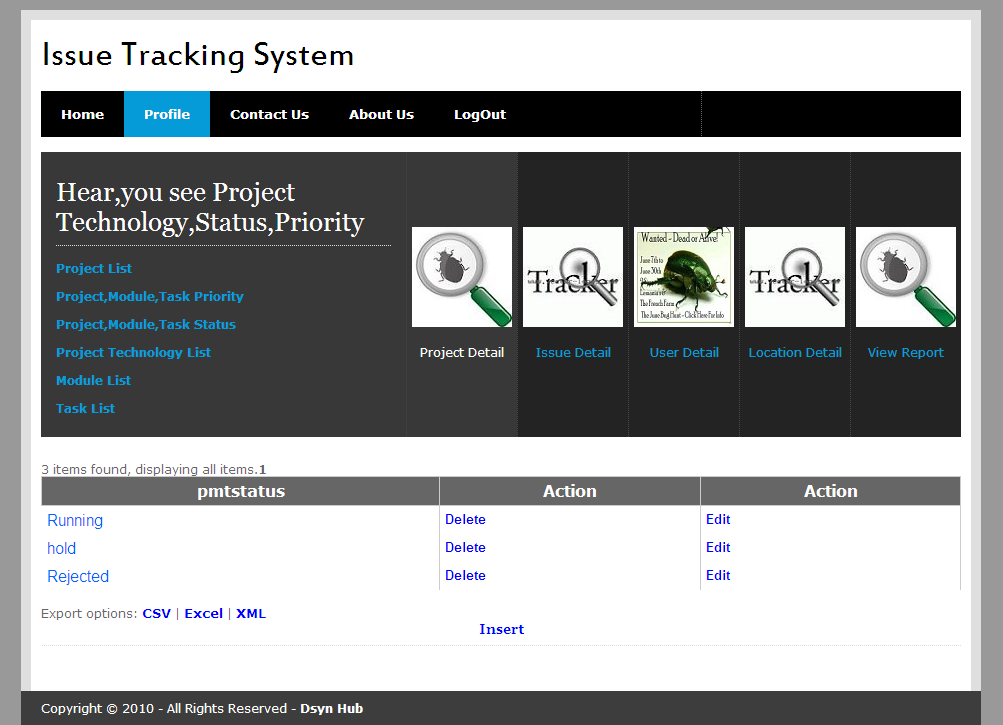
1. **Security Question List Form:-**



1. **Operating System Insert:-**

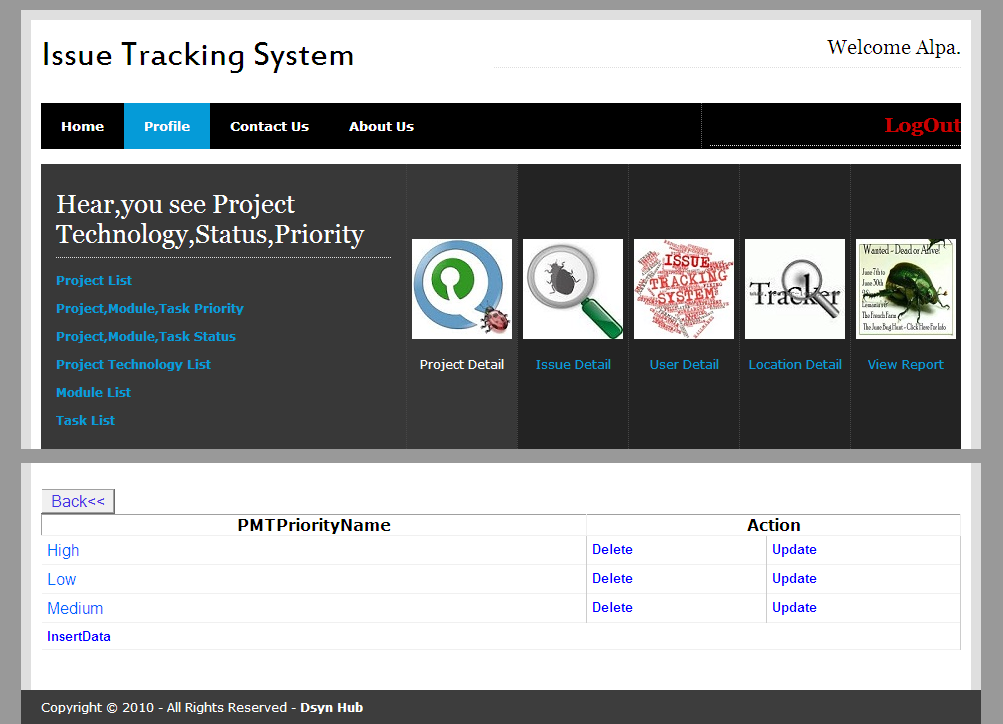


1. **Project,Module,Task Status List:-**



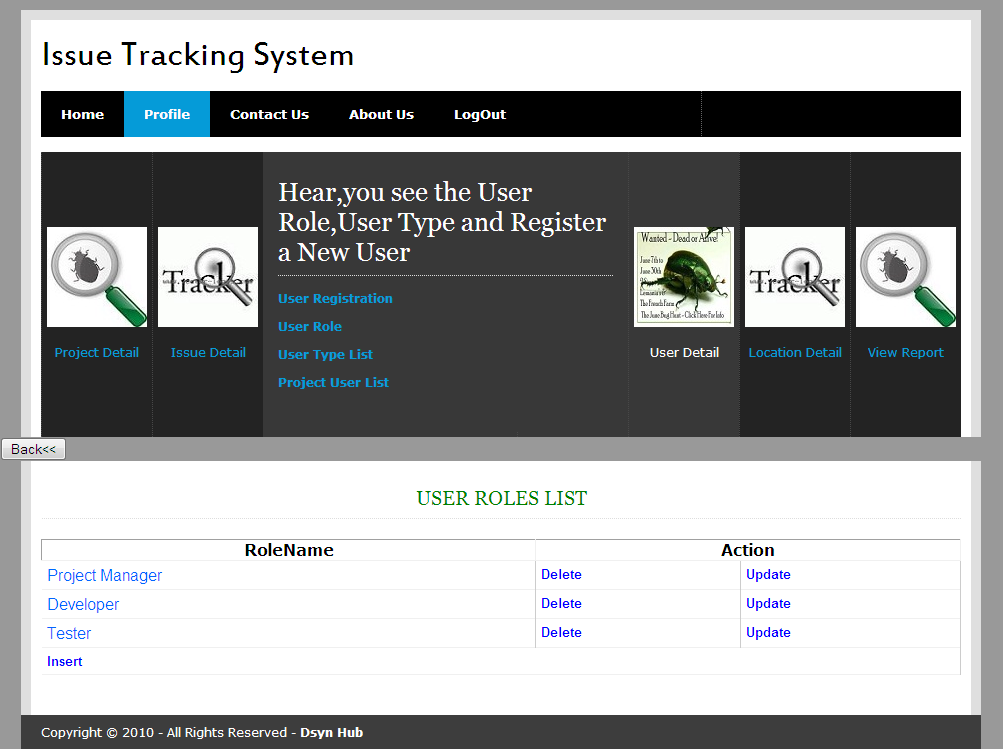
List of project , module and task Status.

1. **Project,Module,Task Priority List:-**



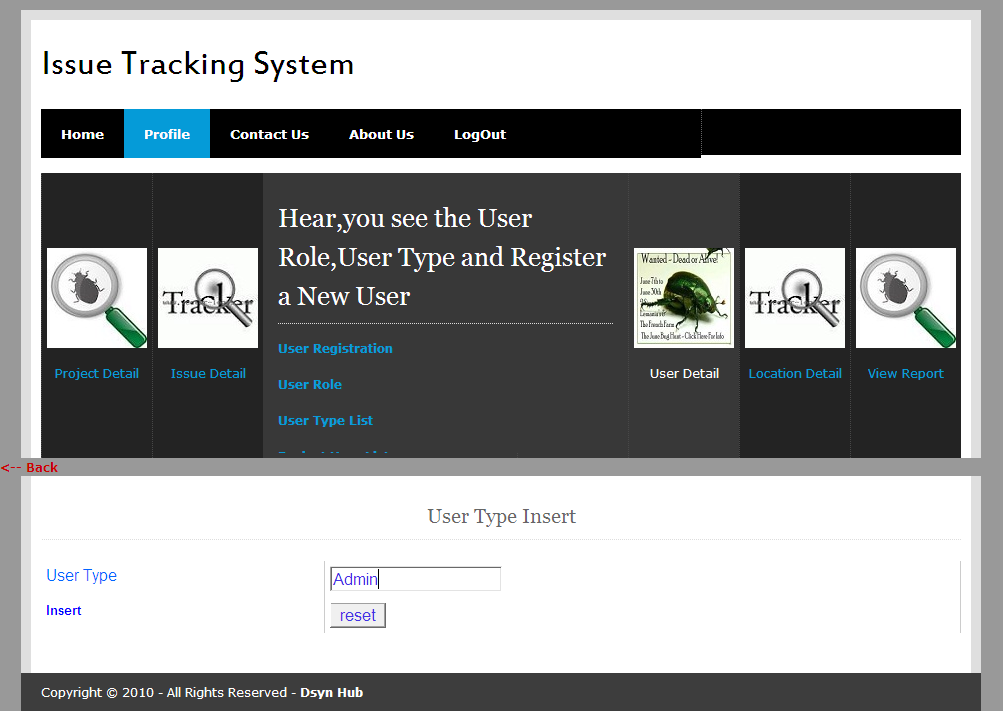
Admin Can Add a project,Module and Task’Priority.

1. **User Role:-**



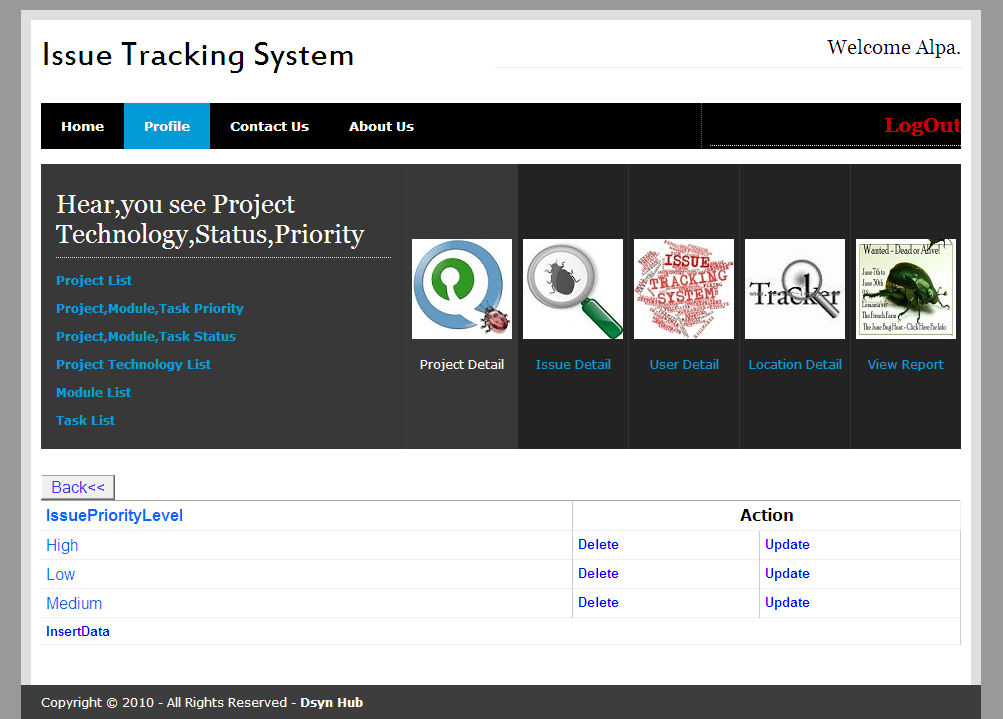
Add will Add User Role. Hear No of user roles are listed .

1. **User Type Insert:-**



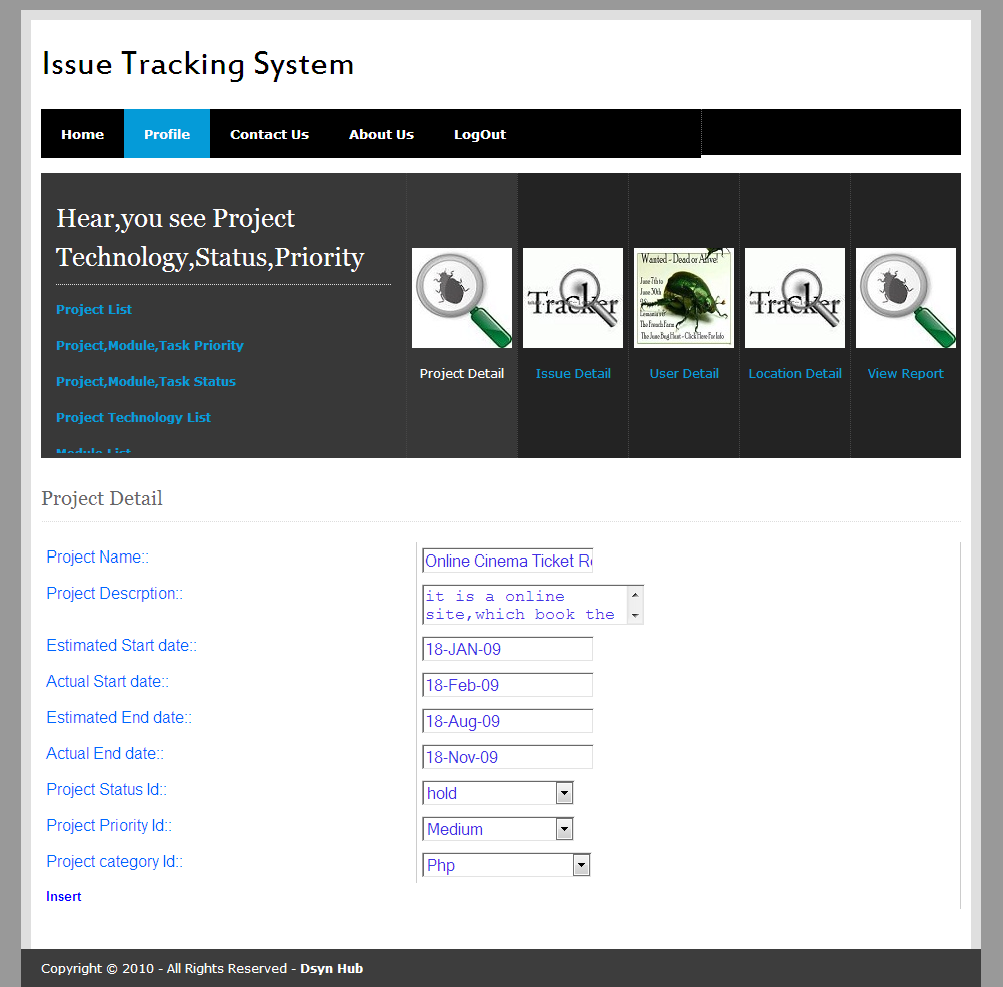
Add User Type.It may be admin or user.

1. **Issue Priority List:-**



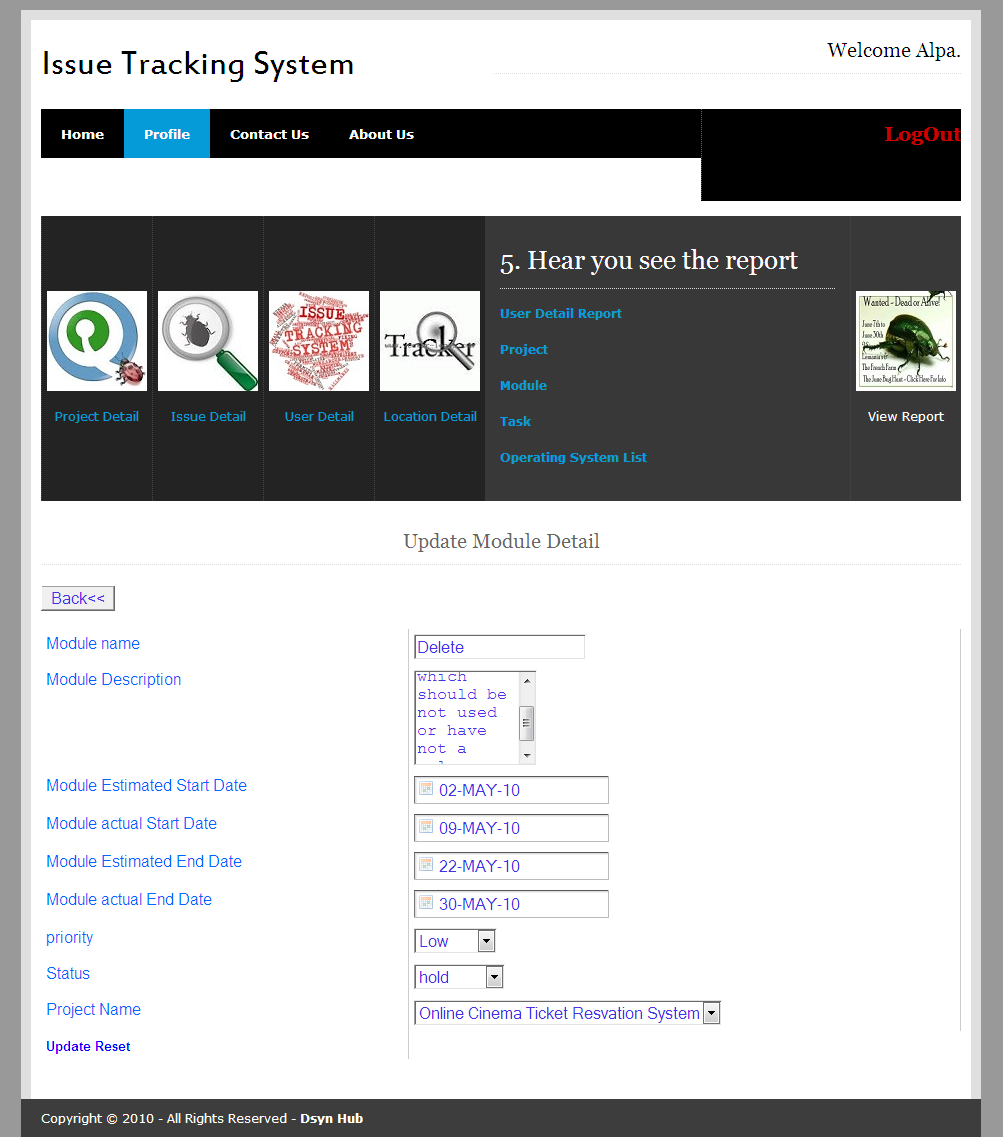
This form List issue Priority.

1. **Project Master Insert:-**



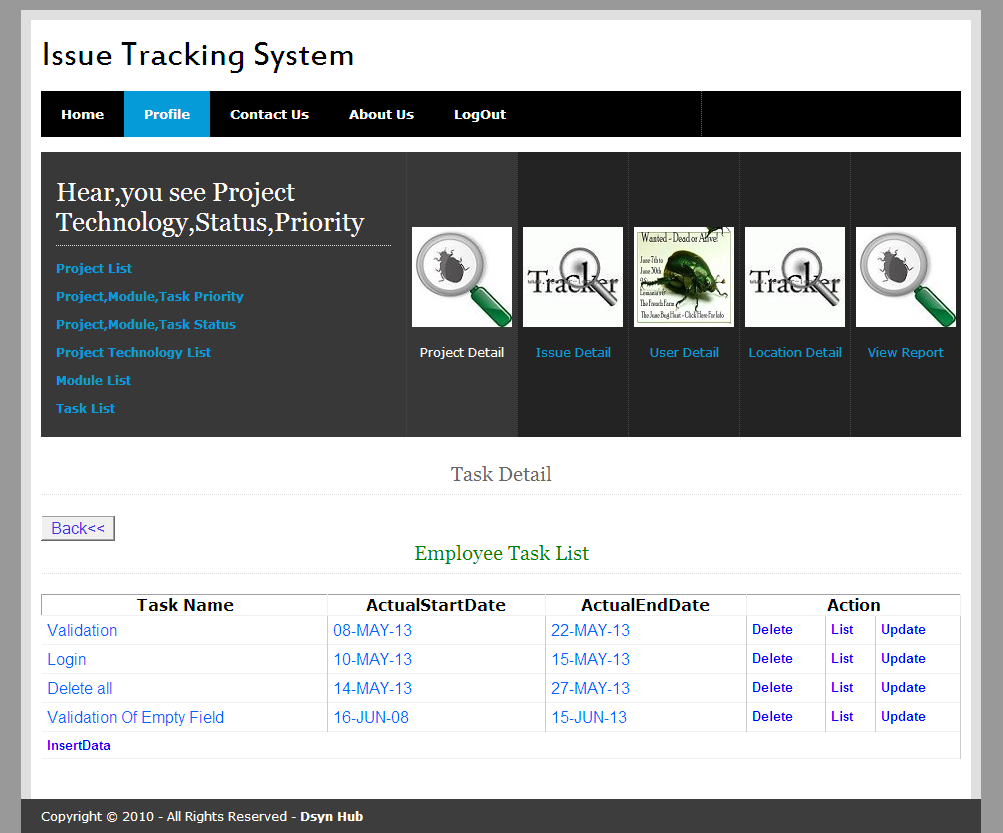
Admin and Project manager are add the project in the system.this form will help to add project.it will contain project status,project priority,project technology.project start and end date.

1. **Module Detail Update:-**



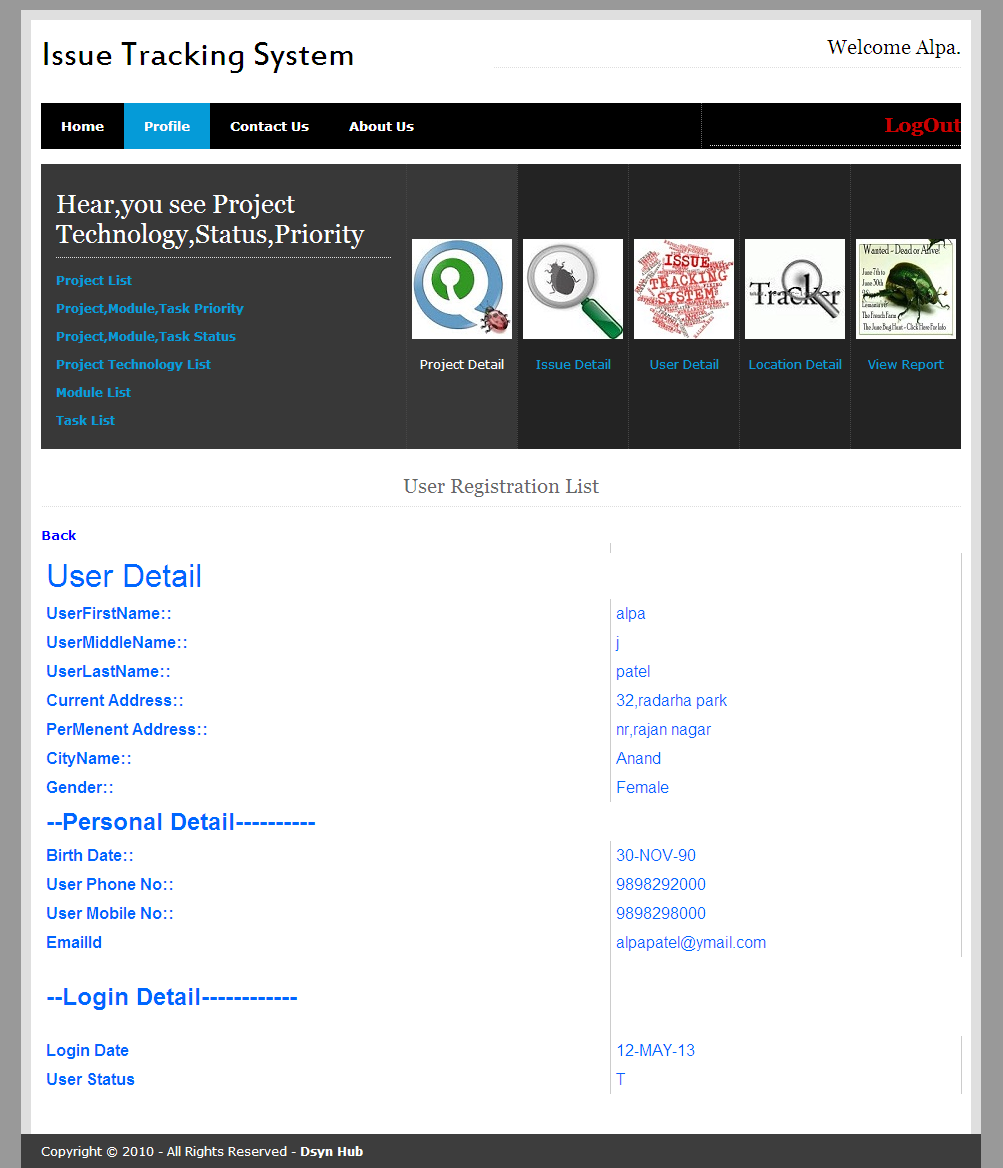
With help of this form user can change or update the details of module.

1. **Task List:-**



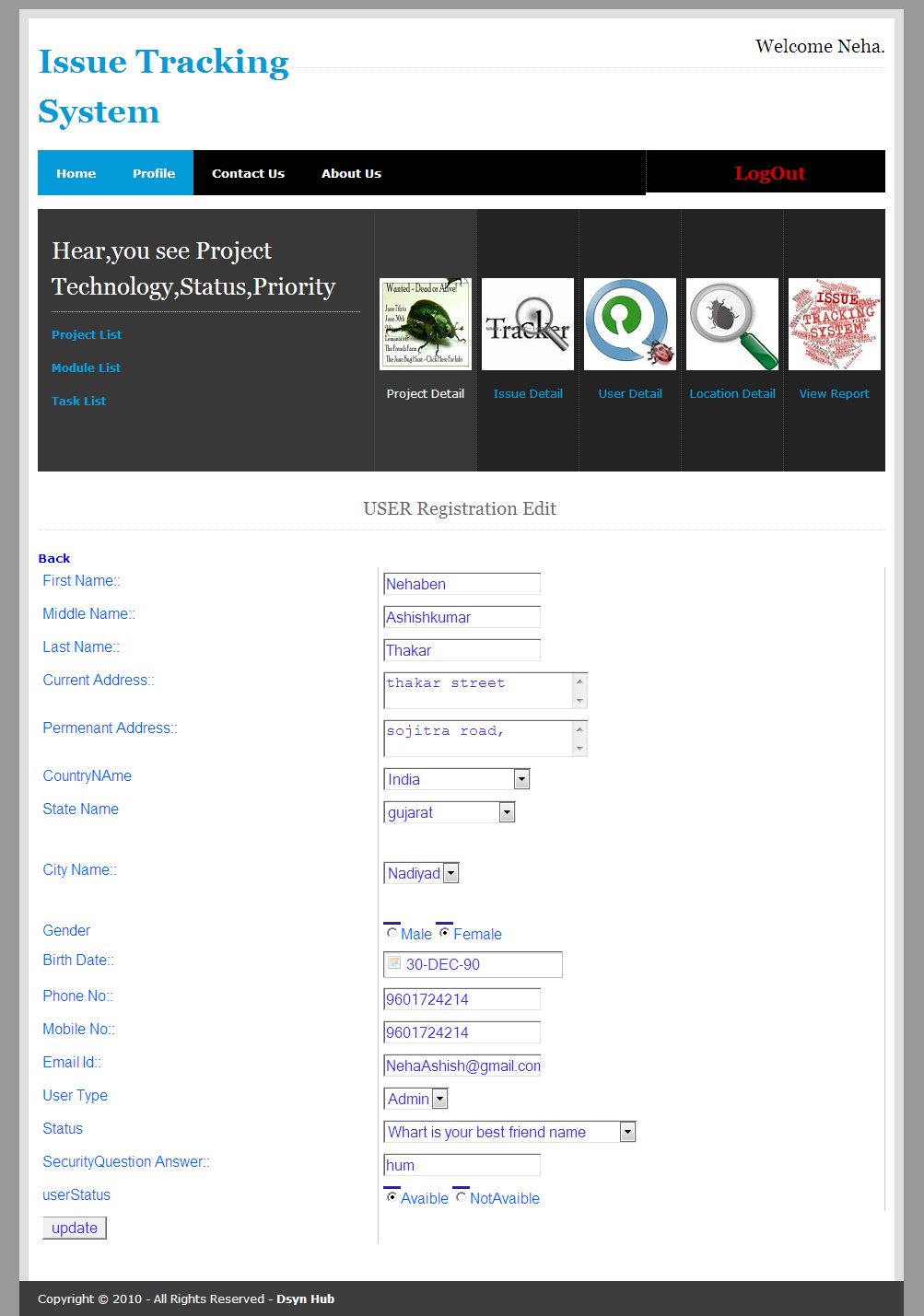
This form gives the details of task of project.

1. **View User Profile :-**



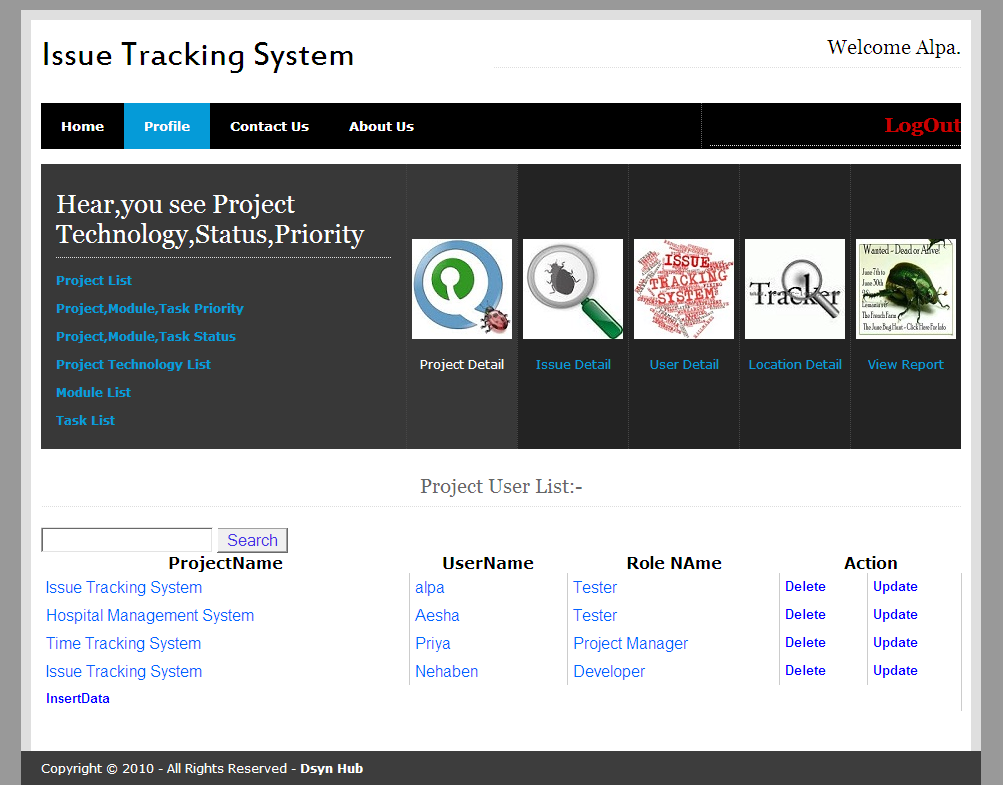
User can View his own Profile.

1. **User Edit Profile:-**



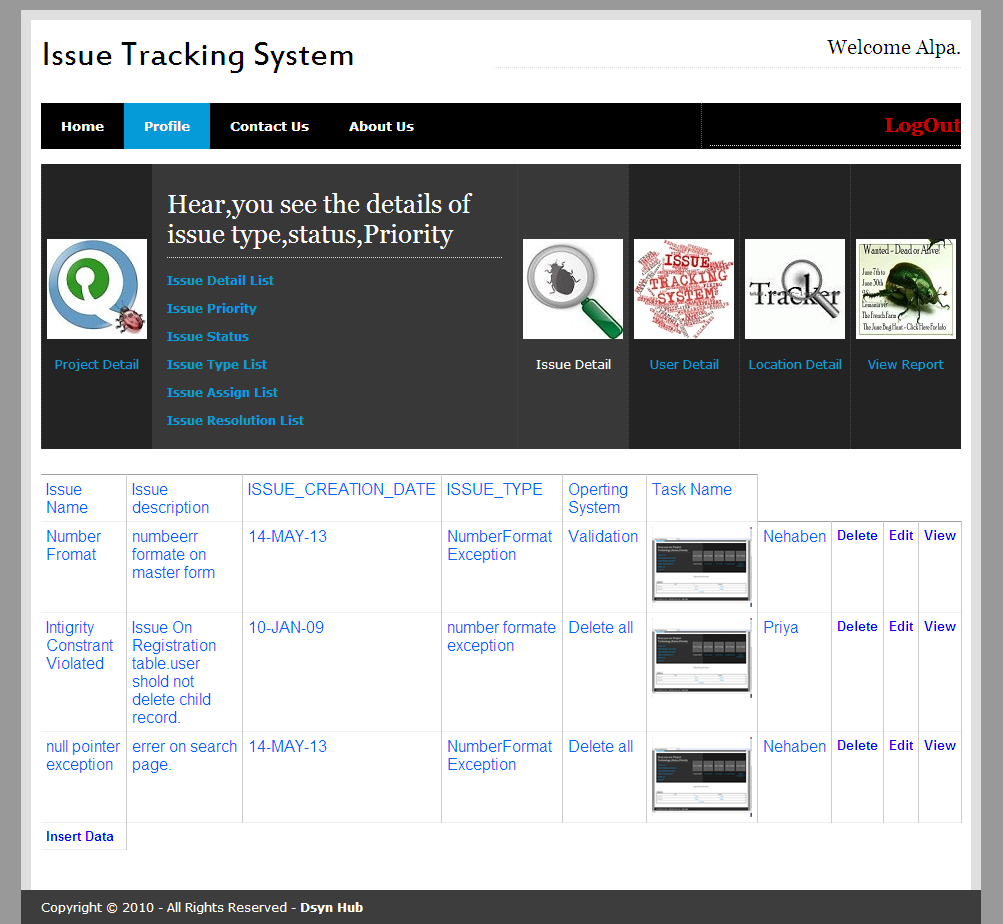
user who can login ,he/she can change his/her profile.

1. **Project User List:-**



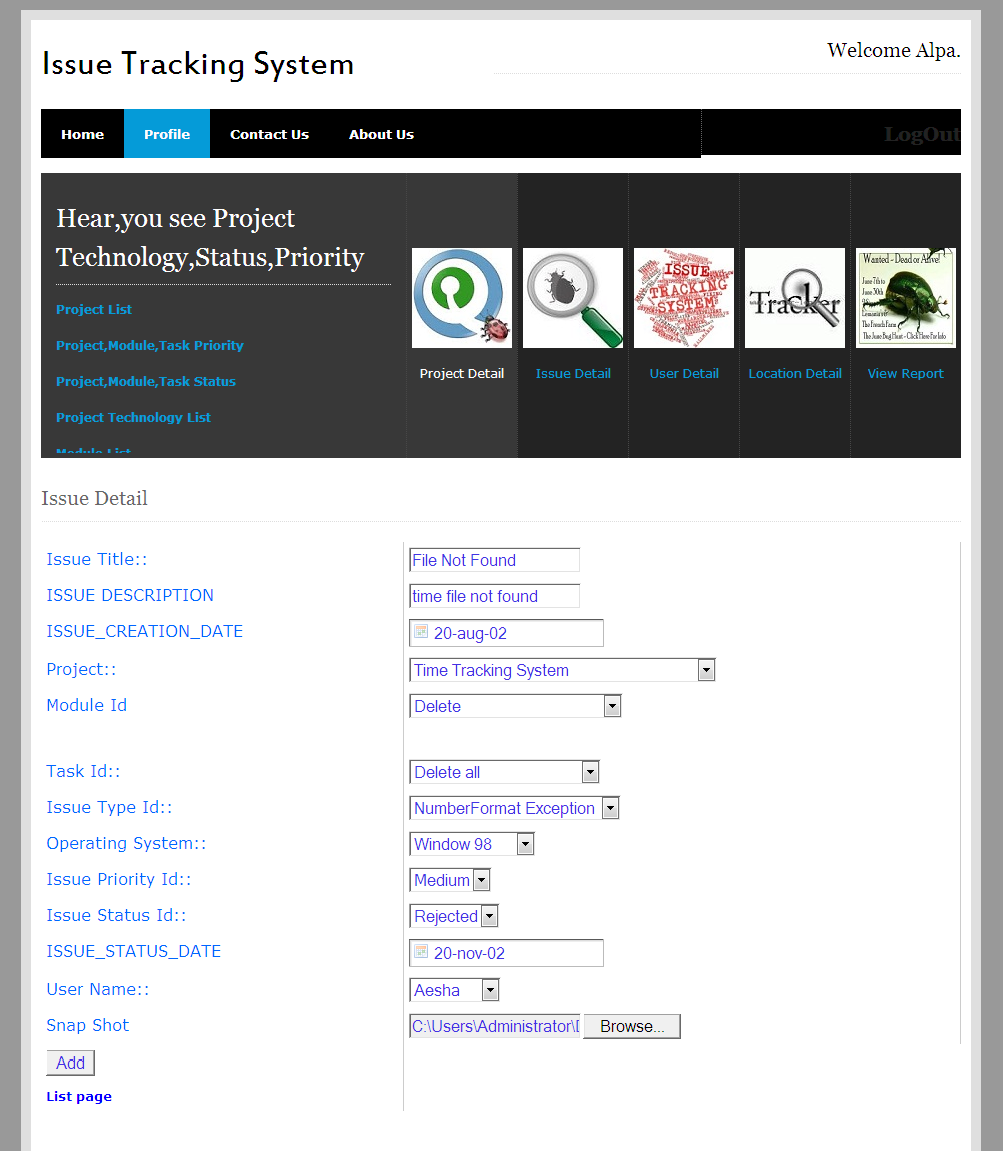
This form disply the details of who can work on which project and also give the detail about user role.

1. **issue Detail List:-**



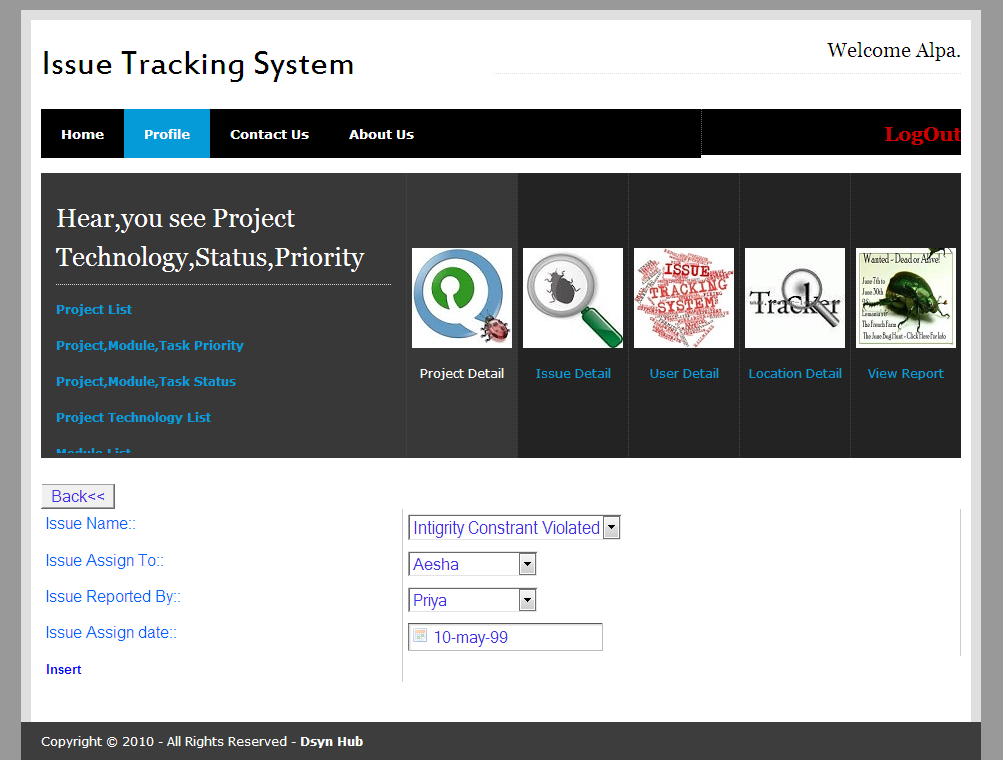
This form gives the details about issue .give the name of issue,date of issue creation etc..

1. **Issue Detail Insert:-**



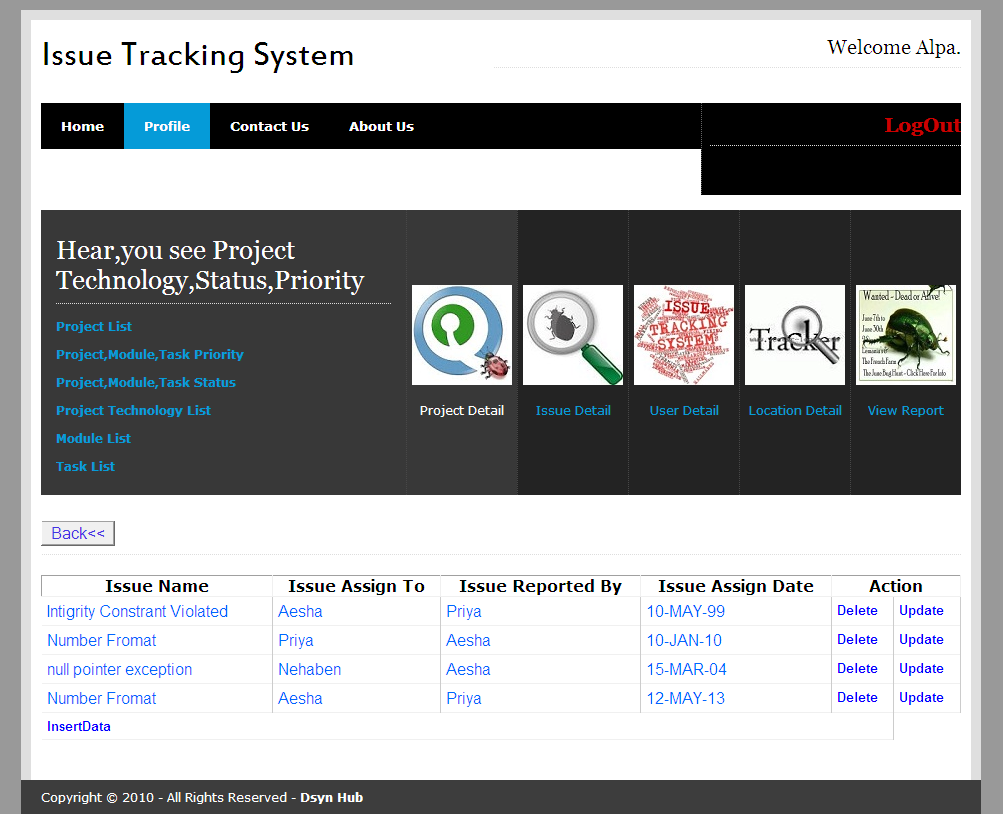
Admin, Project manager ,tester should add the issue.

1. **Issue Assign Insert:-**



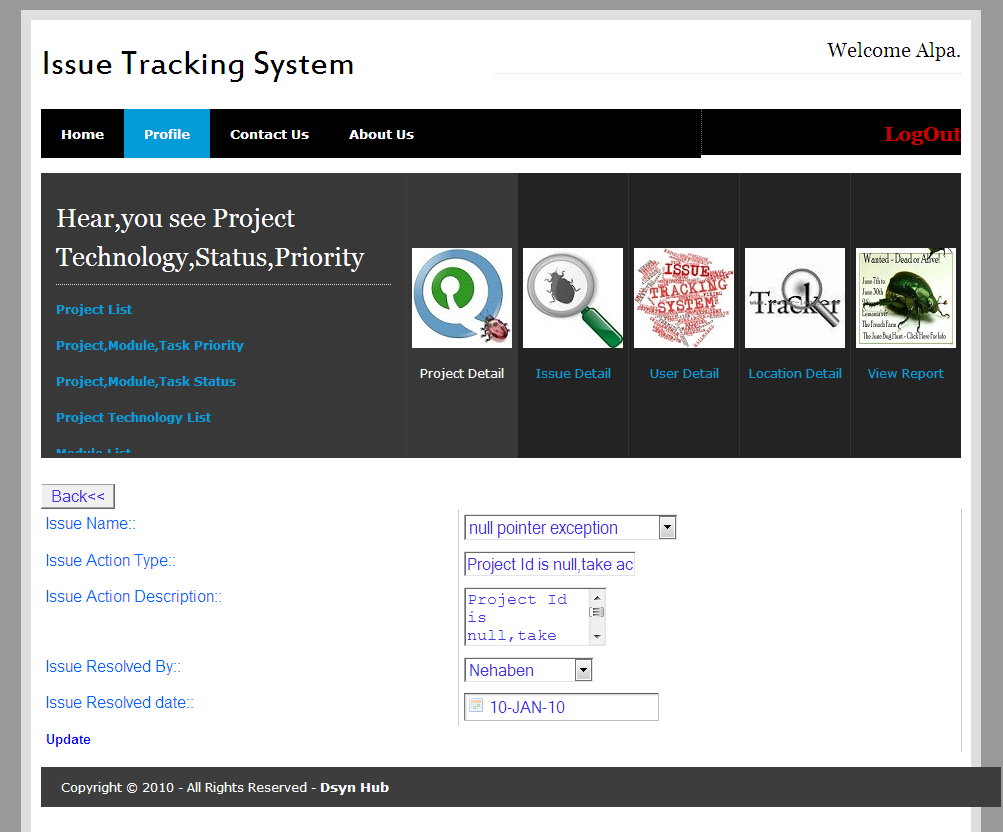
Assign a issue to the user.

1. **Issue Assign List:-**



Give the details about issue assign. Give the information about issue assign by and issue assign by,

1. **Issue Resolved Update**:-

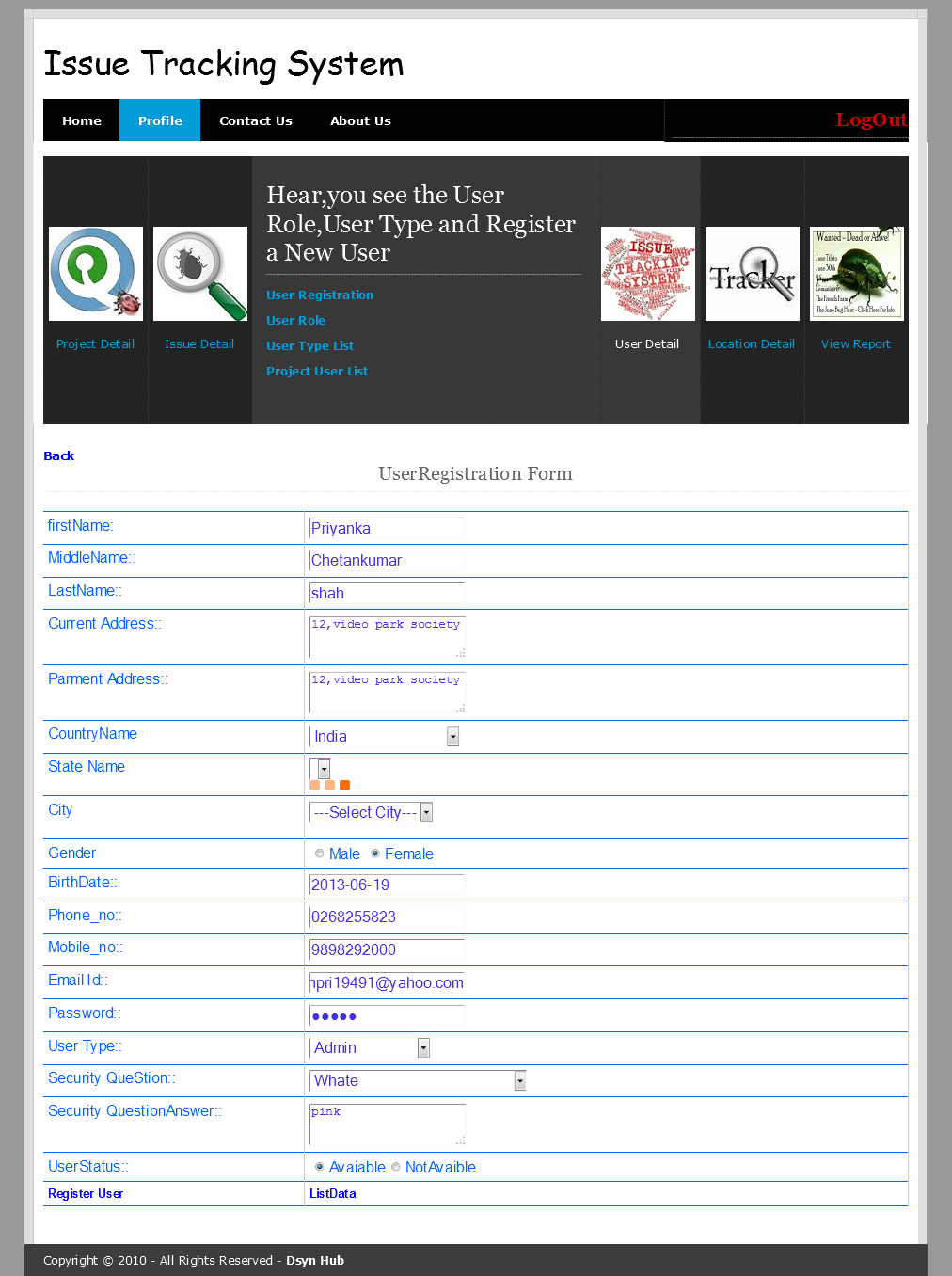


Update the details about issue resolved.

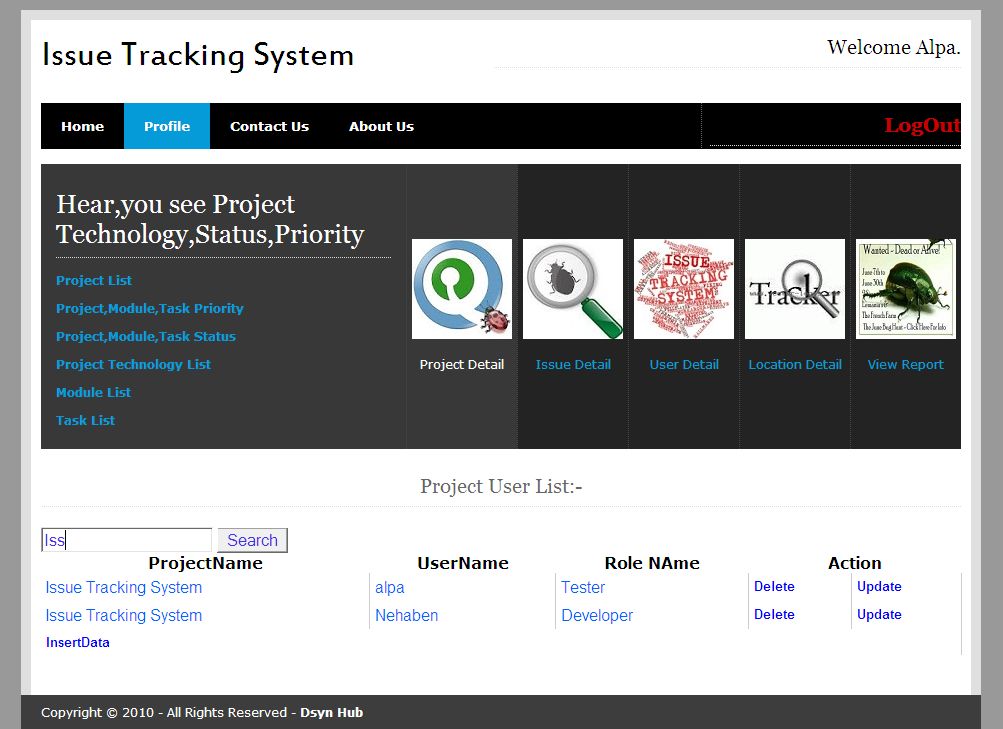
**Functinality:-**

**Directing Web Remoting(DWR)**

1. **(City name Comes from State name and State name Come From Country name):-**



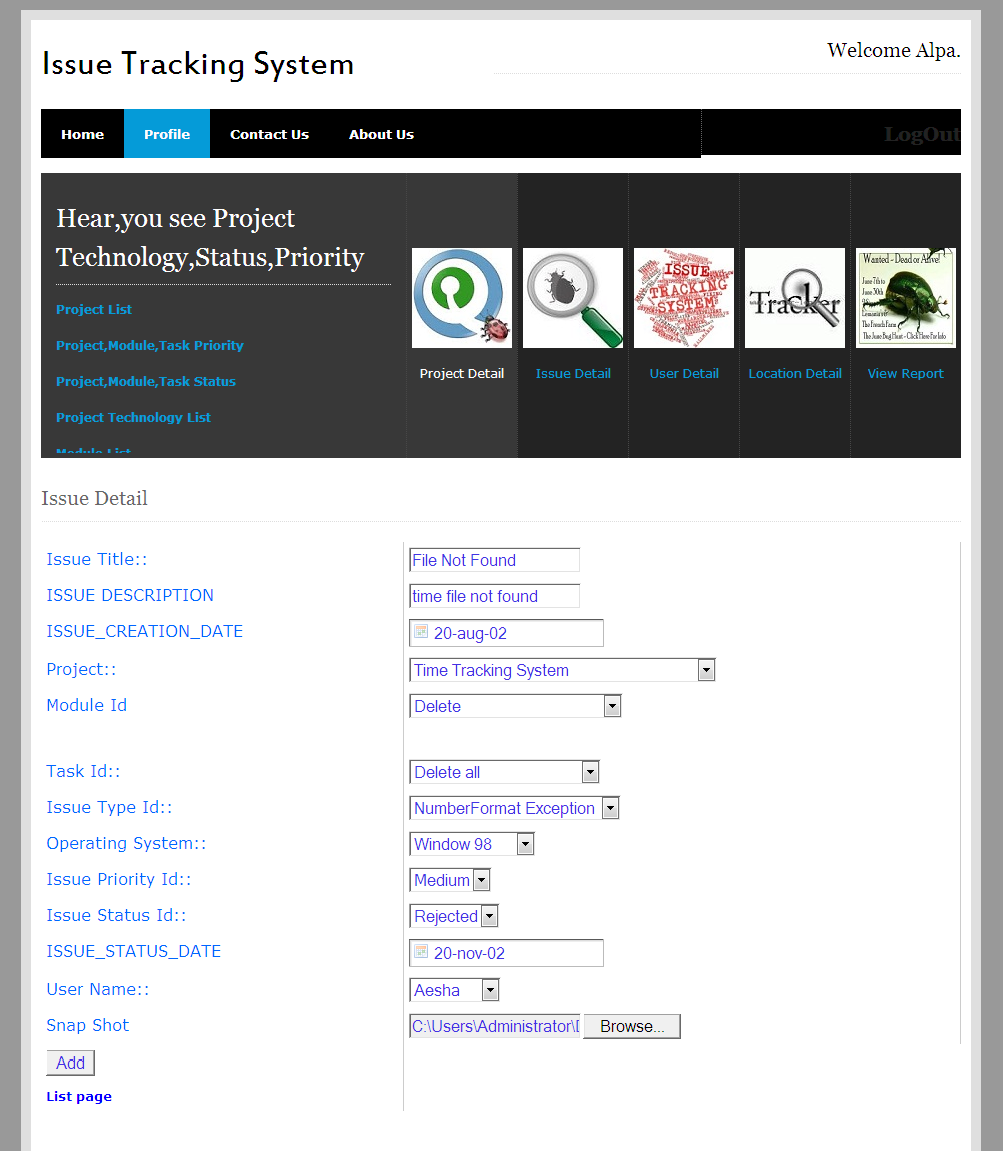
1. **Search Form:-**



User Can Search by whole name of user,project.he featch details about project name,who is work on the project and what is the role of user.

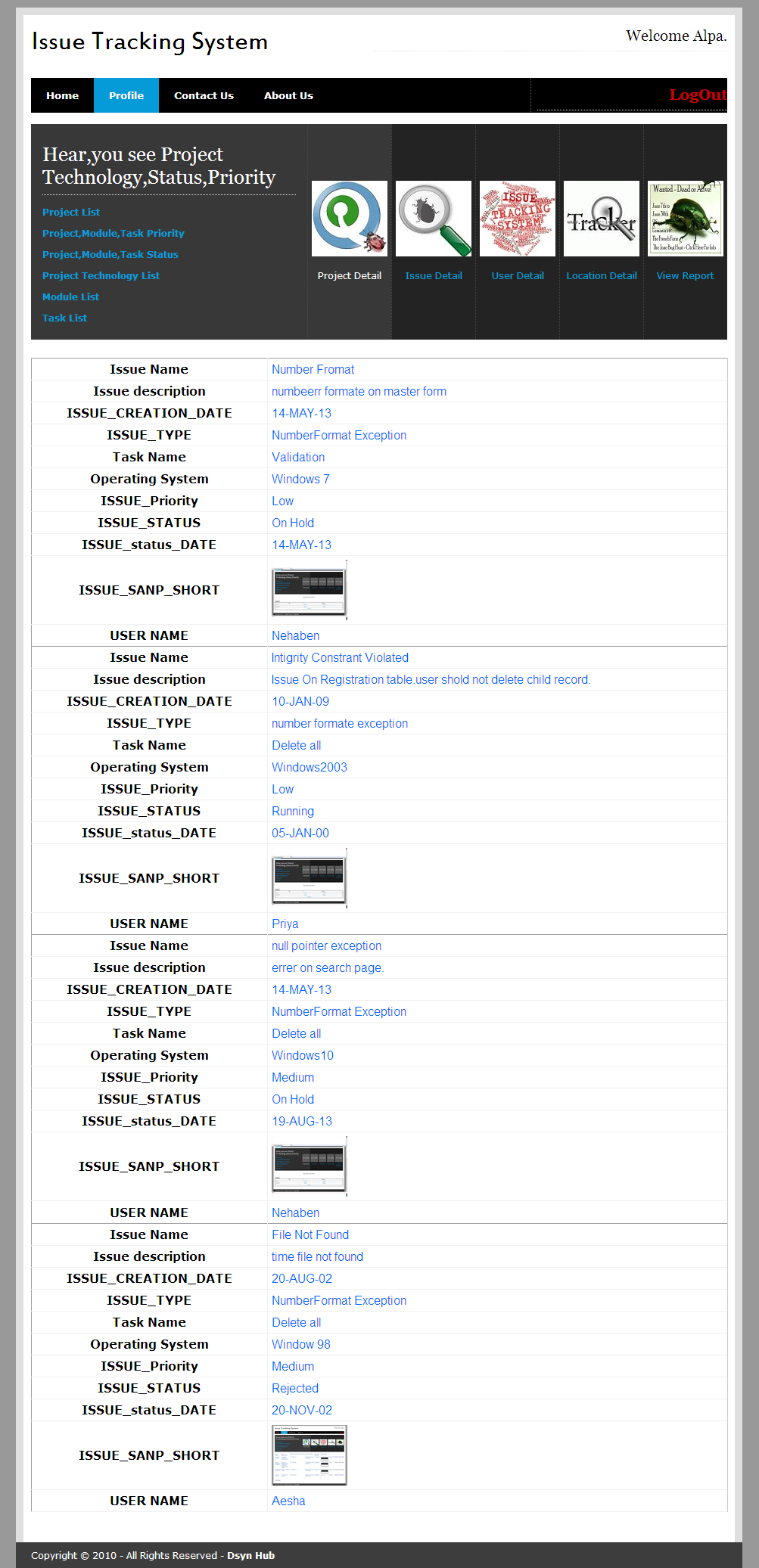
User can also find details by writing the first letter of user name or project.

1. **File Upload:-**



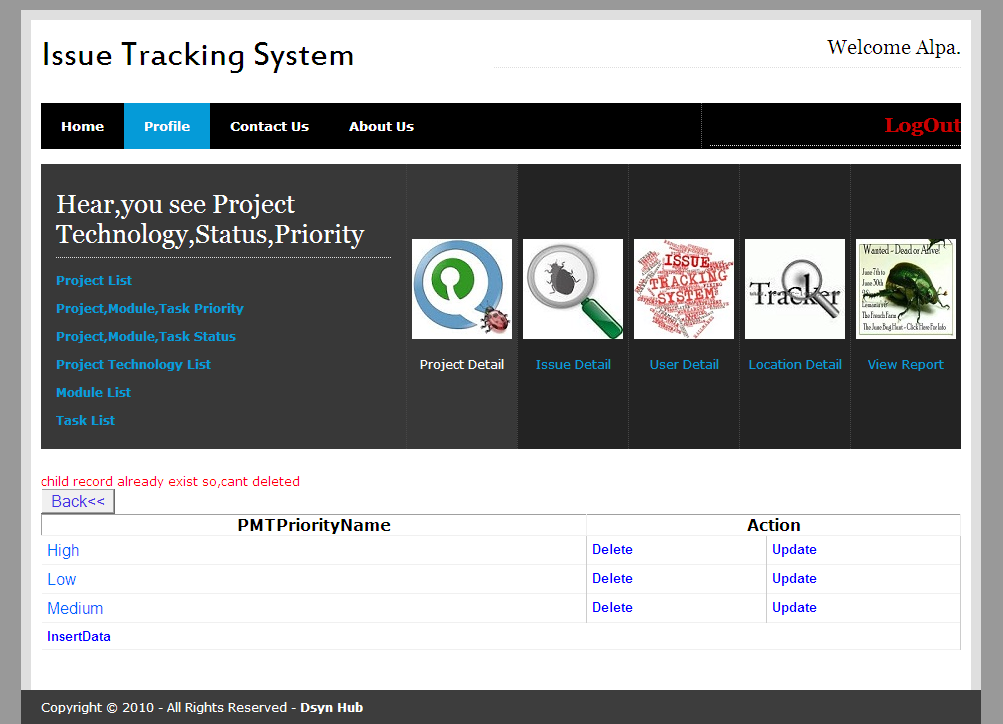
User can upload a snap short of error page.and also display the snap short.

1. **View the Upload File**



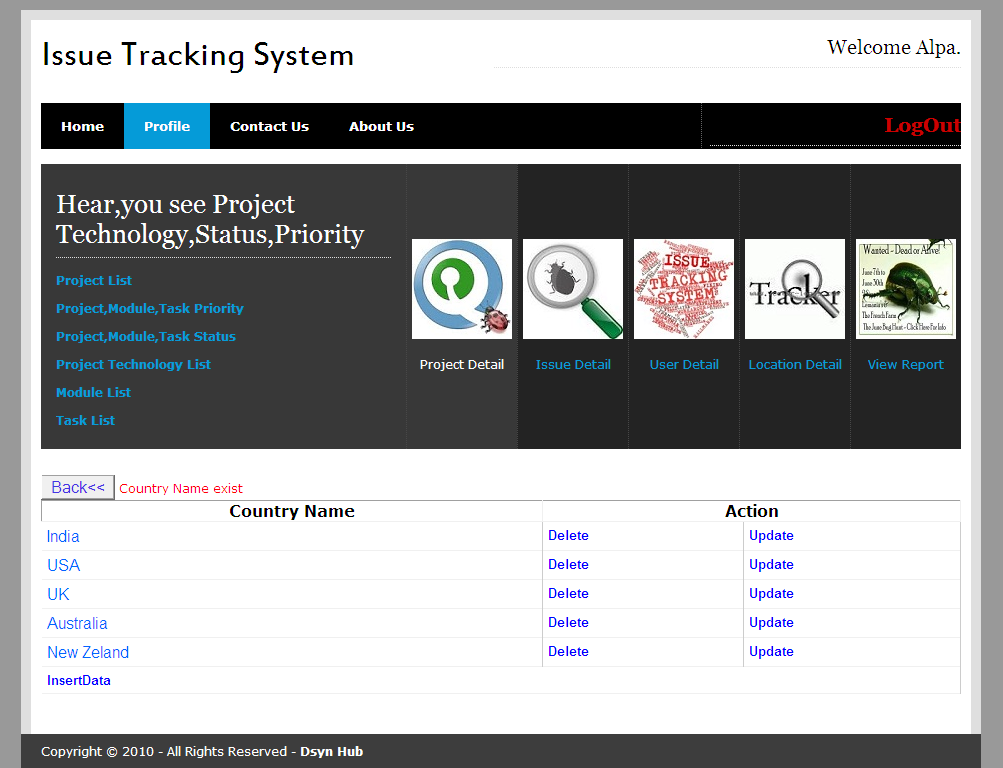
Display Data in Cvv,xml or excel file:-

1. **Delete(Child record not deleted):-**



Hear high is child record,so it can not be deleted..

1. **Insert(Duplicate Value Not Inserted):-**



Chapter: 8

Testing

**TESTING**

**8.1 Testing Plan**

**8.2 Testing Strategy**

**8.3 Testing Methods**

**8.4 Test Cases**

Project Testing is the process of executing a program or system with the intent of finding errors. It involves any activity aimed at evaluating an attribute or capability of a program or system and determining that it meets its required results.

Testing can be stated as the process of validating and verifying that a software program/application/product meets the requirements that guided its design and development; works as expected; and can be implemented with the same characteristics.

Testing can be implemented at any time in the development process. However, most of the test effort traditionally occurs after the requirements have been defined and the coding process has been completed having been shown that fixing a bug is less expensive when found earlier in the development process.

**8.1 TESTING PLAN:**

A test plan documents the strategy that will be used to verify and ensure that a product or system meets its design specifications and other requirements. A test plan is usually prepared by or with significant input from [Test Engineers](http://en.wikipedia.org/wiki/Test_Engineer). Depending on the product and the responsibility of the organization to which the test plan applies, a test plan may include one or more of the following:

* Design Verification or Compliance test - to be performed during the development or approval stages of the product, typically on a small sample of units.
* Manufacturing or Production test - to be performed during preparation or assembly of the product in an ongoing manner for purposes of performance verification and quality control.
* Acceptance or Commissioning test - to be performed at the time of delivery or installation of the product.
* Service and Repair test - to be performed as required over the service life of the product.
* Regression test - to be performed on an existing operational product, to verify that existing functionality didn't get broken when other aspects of the environment are changed (e.g., upgrading the platform on which an existing application runs).

A complex system may have a high level test plan to address the overall requirements and supporting test plans to address the design details of subsystems and components. Test plan document formats can be as varied as the products and organizations to which they apply. There are three major elements that should be described in the test plan: Test Coverage, Test Methods, and Test Responsibilities. These are also used in a formal [test strategy](http://en.wikipedia.org/wiki/Test_strategy).

**Table 8.1 Types Of Testing**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr.No.** | **Type of Testing** | **Responsibility** | **Remarks** |
| 1 | **Unit testing** | Independent code (working in any circumstances) | Checking whether the code is  Flexible for further changes to be made. Will it affect any other module to any extent or not. Perform alpha testing to reduce bugs. N-Unit tool used. |
| 2 | **Integration** | Testing done to check  Interdependencies of units. No gaps in data flow | To ensure that running the  Current test affects the previous code to achieve each test case. |
| 3 | **System** | System as a whole working on any environment |  |
| 4 | **Acceptance** | System should operate in the  Manner expected. End user  Maps his initial requirements with the system. | To ensure no gaps. To achieve the overall functionality |

**8.2 TESTING STRATGIES:**

Local data are examined to ensure that integrity is maintained. Boundary conditions are tested. Basis path testing should be used. All error handling paths should be tested. Drivers and/or stubs need to be developed to test incomplete software.

* **INTEGRATION TESTING:**

1. **TOP-DOWN INTEGRATION TESTING:**

Main control module used as a test driver and stubs are substitutes for components directly subordinate to it. Subordinate stubs are replaced one at a time with real components (following the depth-first or breadth-first approach). Tests are conducted as each component is integrated. On completion of each set of tests and other stub is replaced with a real component. Regression testing may be used to ensure that new errors not introduced.

1. **BOTTOM-UP INTEGRATION TESTING:**

Low level components are combined in clusters that perform a specific software function. A driver (control program) is written to coordinate test case input and output. The cluster is tested. Drivers are removed and clusters are combined moving upward in the program structure.

**8.3 TESTING METHODS:**

Testing presents interesting anomaly for the software engineering activities, the engineer attempts to build software from an abstract concept to a tangible product. Now comes testing. The engineer creates a series of test case that are initiated to “demolish” the software that has been build.

* **MODELS OF TESTING:**

There are different Models of testing. On the basis of testing methods there are two types of testing:

1. Black-box testing

2. White-box testing

* **BLACK BOX TESTING:**

1. **EQUIVALENCE PARTIONING:**

Black-box technique that divides the input domain into classes of data from which test cases can be derived an ideal test case uncovers a class of errors that might require many arbitrary test cases to be executed before a general error is observed. Equivalence class guidelines:

* If input condition specifies a range, one valid and two invalid equivalence classes are defined.
* If an input condition requires a specific value, one valid and two invalid equivalence classes are defined.
* If an input condition specifies a member of a set, one valid and one invalid equivalence class is defined.
* If an input condition is Boolean, one valid and one invalid equivalence class is defined.

1. **BOUNDARY VALUE ANALYSIS:**

Black-box technique that focuses on the boundaries of the input domain rather than its entering.BVA guidelines:

* If input condition specifies a range bounded by values a and b, test cases should include a and b, values just above and just below a and b.
* If an input condition specifies and number of values, test cases should be exercise the minimum and maximum numbers, as well as values just above and just below the minimum and maximum values.
* Apply guidelines 1 and 2 to output conditions, test cases should be designed to produce the minimum and maxim output reports if internal program data structures have boundaries (e.g. size limitations), be certain to test the boundaries.
* **WHITE-BOX TESTING:**

1. **STATEMENT COVERAGE:**

Statement coverage methodology focuses on, designing test cases so that, every statement in a program is executed at least once. No statement in the program should remain unreachable.

1. **BRANCH COVERAGE:**

Test cases are designed such that, different branch conditions. Give true and false values in some execution. All branches are traversed. Branch testing guarantees statement coverage. It is a stronger testing compared to the statement coverage-based testing.

1. **PATH COVERAGE:**

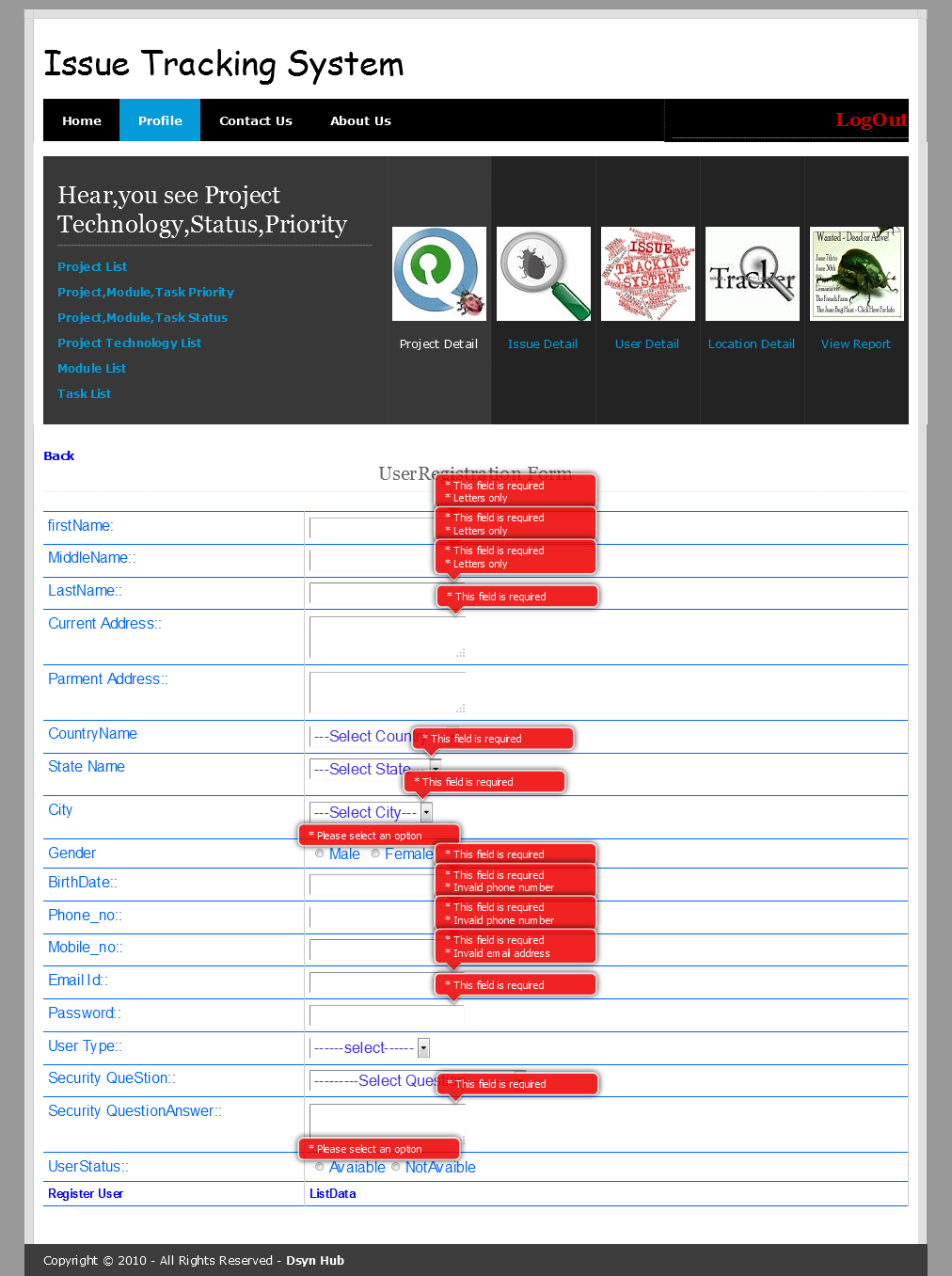
Design test cases such that, all linearly independent paths (LIP) in the program are executed at least once. To understand the path coverage-based testing we need to learn how to draw control flow graph of a program. A control flow graph (CFG) describes the sequence in which different instructions or statements of a program get executed. The way control flows through the program.

1. **CONDITION COVERAGE:**

Test cases are designed such that, each component of a composite conditional expression It help us to, Gives both true and false values. To check for all combination of conditions.

**Validation**

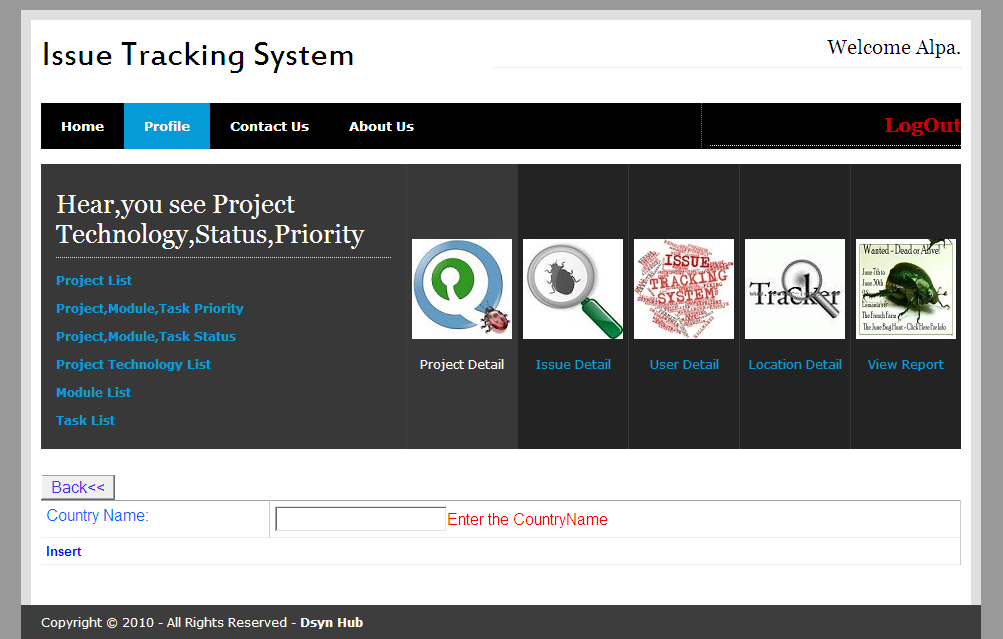
1. **Client Side Validation:-**



It will give the error message on page.if textbox should be blank,phone no should not valid,Email id should be blank or invalid..

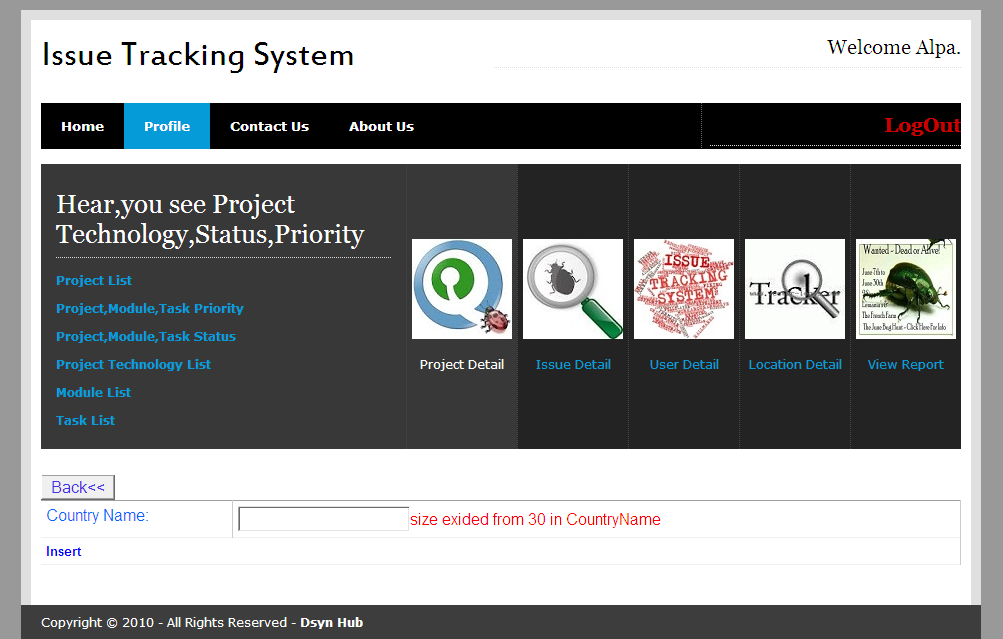
Server Side Validation:-

1. **Blank Not Valid:-**



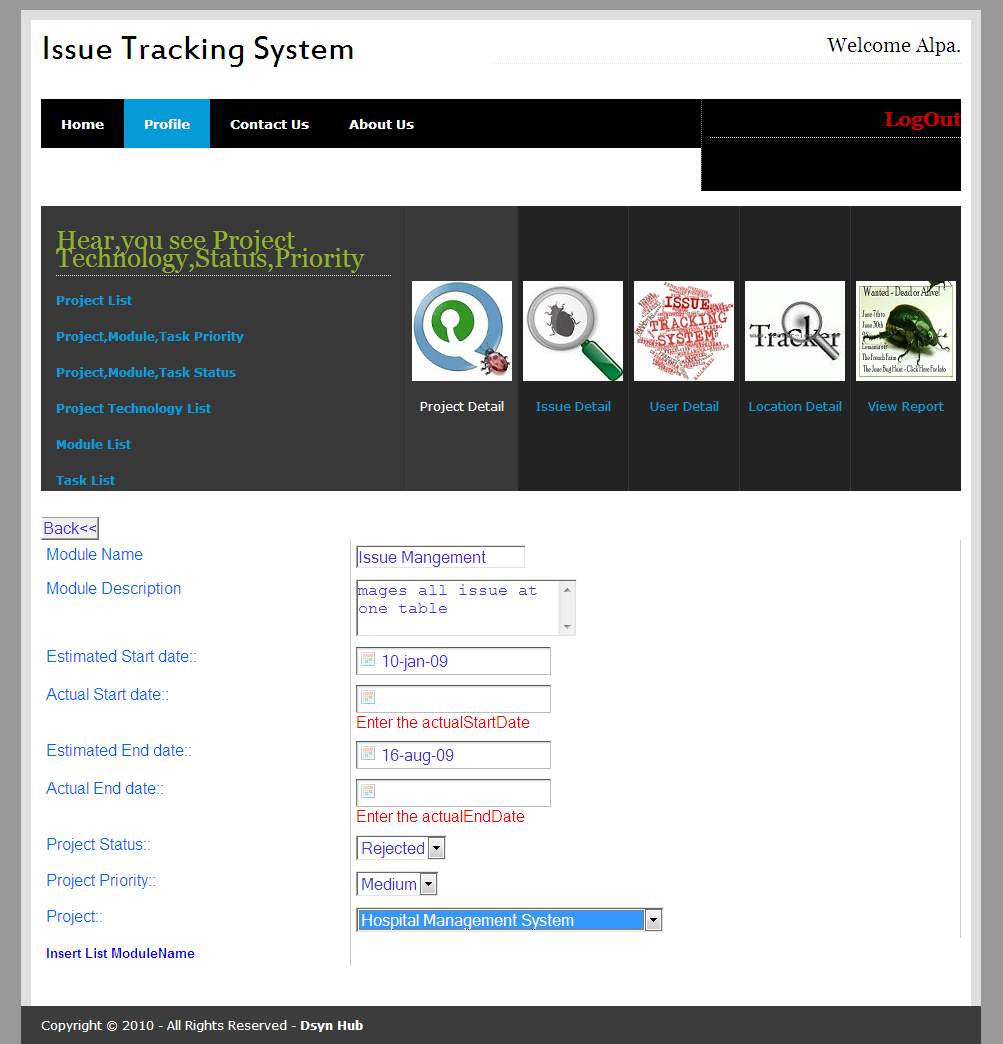
Blank value Should not enter..it will give the error.

1. **Maximum Size:-**



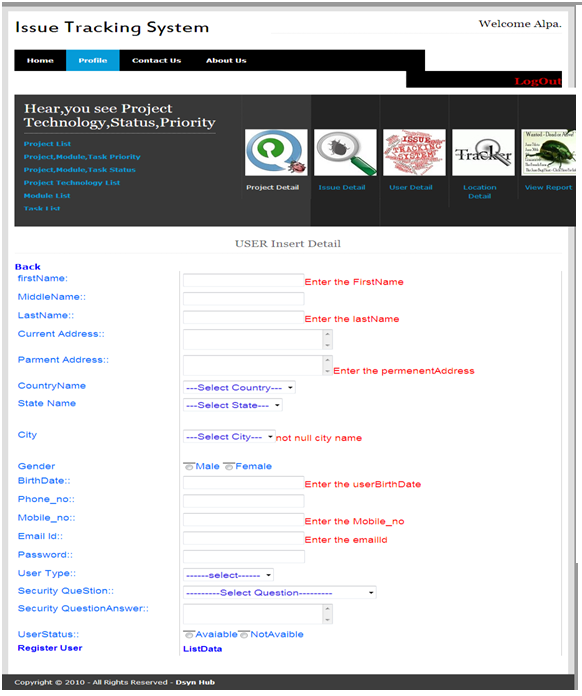
Country name should be less than 30,if yu enter more than 30 charcter he shold give the error.

1. **Date Validation:-**



Give the proper date.

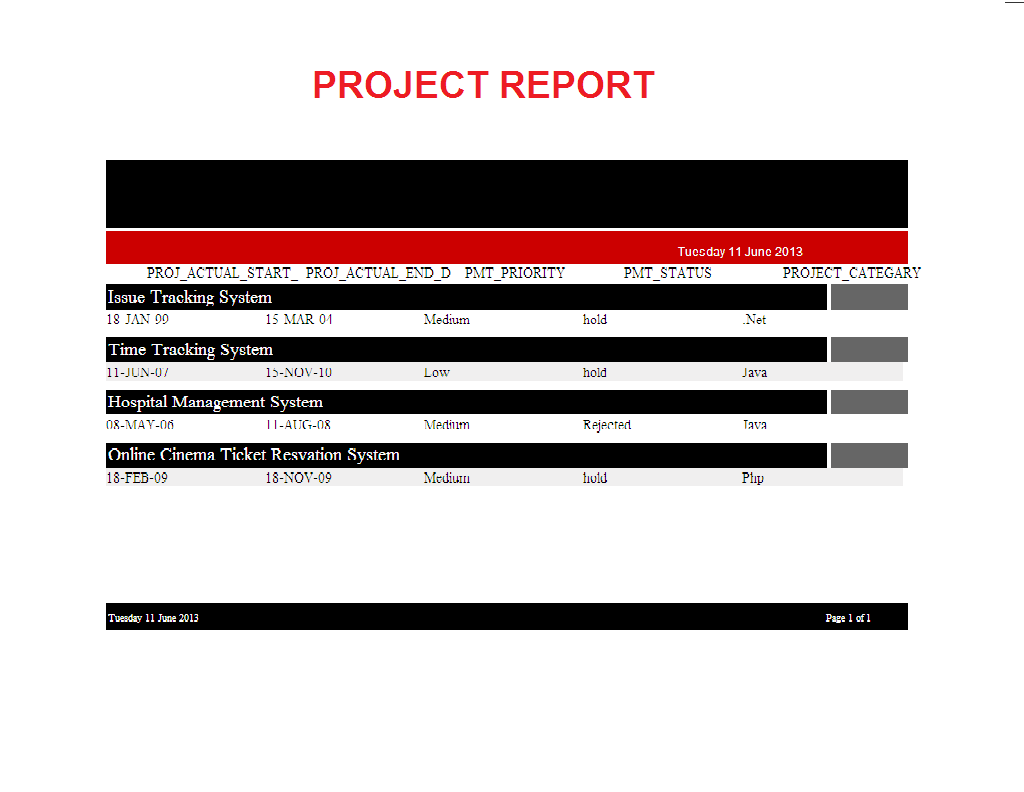
1. **All Validation:-**



In this form all validation are there.

**Jasper Report:-**

1.Project Detail:-

****

**.2.Issue Detail Report:-**

****

Chapter: 9

Future Enhancement

**9.1 Future Work**

* I want to improve our Application, as it is the more user friendly and simple, Main thing which attracts all users.
* We Would To Make The Site More Eyes Catching As Compared To Current Page. We Can Use Flash So That The Site Becomes More Dynamic. Obviously Good on the Project More In Terms of Functionality. Since Functionality is The Key to Good Projects.
* I will provide SMS feature for informing Client for Solved issue.
* In future, we will work on mailing feature.
* Through this feature, we can send mail to the Issue Tracker.
* I would to make Scheduler which keeps Record of all Issue, such as when it finished, its last date of solve issue.
* I would to make upload download module for uploading error page.

CONLUSION AND DISSCUSSION

**Conclusion**

After having study of entire system we came to know that the given system is fulfilling the total requirements and also working very efficiently along with higher performance. Even the system has be developed in such a manner so that the user could be easily be familiar with the system with a small training session about application. The system has been developed using platform java which provides higher stability to the time tracking system.

There is a good scope to have better development in future in the same system without whole system replacement. Any system developed with such a good concept is always acceptable not only in the current scenario but also for the future development.

Chapter: 10

Bibliography

**Books**

During the development of software we used following books for reference:

The Complete Reference Java2

**-** By Herbert Schildt

Object Oriented Analysis Design

**-** By Rambuag

Software Engineering

**-** Ian Sommerville

**Online References**

[**www.dreamingsoft.com**](http://www.dreamingsoft.com)

[**http://www.oracle.com/technetwork/database/database10g/overview/index.html**](http://www.oracle.com/technetwork/database/database10g/overview/index.html)

[**http://www.oracle-base.com/articles/misc/apache-tomcat-7-installation-on-windows.php**](http://www.oracle-base.com/articles/misc/apache-tomcat-7-installation-on-windows.php)

[**http://directwebremoting.org/dwr/index.html**](http://directwebremoting.org/dwr/index.html)