2010

Design and Functional Specification

1. Introduction

1.1 Purpose

This document records functional specifications for Science Technology English Math (STEM) web application along with the high level design. The objective is to document the specifications in as much detail as possible so that the developers may directly take this as a basis for the coding. This document provides an overview of each specific module of the application and technical design, including a description of each component's high level role. Where applicable, the document provides references to more detailed external documentation.

1.2 Scope

This document covers the high level design and functional specification for student, Teacher and Admin modules in STEM.
Some of the web pages will also be developed with minimum content to enable faster loading of pages for low bandwidth users. Design for these web pages for low bandwidth in not included in this document. This will be taken care of at the development time.

2 Technology stack

STEM web site will be developed using following technologies:

Name	Description
ASP.Net 3.5	ASP.NET is a powerful platform for building dynamic web applications that provides a tremendous amount of flexibility and power for building just about any kind of web application from small, personal websites through to large, enterprise-class web applications.
Net Framework 3.5	The .NET Framework is an integral Windows component that supports building and running the next generation of applications and Web services. The key components of the .NET Framework are the common language runtime (CLR) and the .NET Framework class library, which includes ADO.NET, ASP.NET, and Windows Forms. The .NET Framework also consists of technologies such as Windows Workflow Foundation (WF), Windows Communication Foundation (WCF), Windows Card Space ,WPF
Microsoft Visual Studio Team System 2008	Microsoft Visual Studio Team System 2008 is an integrated application lifecycle management (ALM) product family with the tools and processes that

	help development teams work more effectively together.
SQL Server 2005	Microsoft SQL Server 2005 is a relational database management system (RDBMS) produced by Microsoft. Its primary query language is Transact-SQL, an implementation of the ANSI/ISO standard Structured Query Language (SQL).
Windows Server 2008	Windows Server 2008 is the most recent release of Microsoft Windows's server line of operating systems.
MVC ASP.Net 1.0	ASP.NET MVC 1.0 provides a new Model-View-Controller (MVC) framework on top of the existing ASP.NET 3.5 runtime. This means that developers can take advantage of the MVC design patterns to create their Web Applications which includes the ability to achieve and maintain a clear separation of concerns (the UI or view from the business and application logic and backend data), as well as facilitate test driven development (TDD).

3 High level Design

At the highest and most abstract level, the logical architecture view of STEM application can be considered to be a set of cooperating services grouped into the following layers, as shown in figure below:
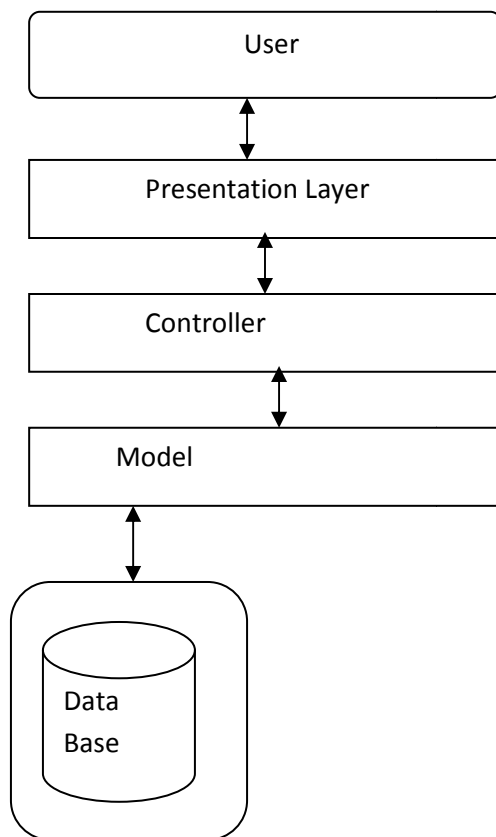
```
        ┌─────────────────────────────┐
        │            User             │
        └─────────────────────────────┘
                      ↕
        ┌─────────────────────────────┐
        │      Presentation Layer      │
        └─────────────────────────────┘
                      ↕
        ┌─────────────────────────────┐
        │         Controller          │
        └─────────────────────────────┘
                      ↕
        ┌─────────────────────────────┐
        │            Model            │
        └─────────────────────────────┘
                      ↕
        ╭─────────────────╮
        │    ┌───────┐    │
        │    │ Data  │    │
        │    │ Base  │    │
        │    └───────┘    │
        ╰─────────────────╯
```

Figure: High level logical architecture view of STEM application

Each layer of application contains a series of components that implement the functionality for that layer. These components should be cohesive and loosely coupled to simplify reuse and maintenance. Following figure shows the components to be implemented in each layer.
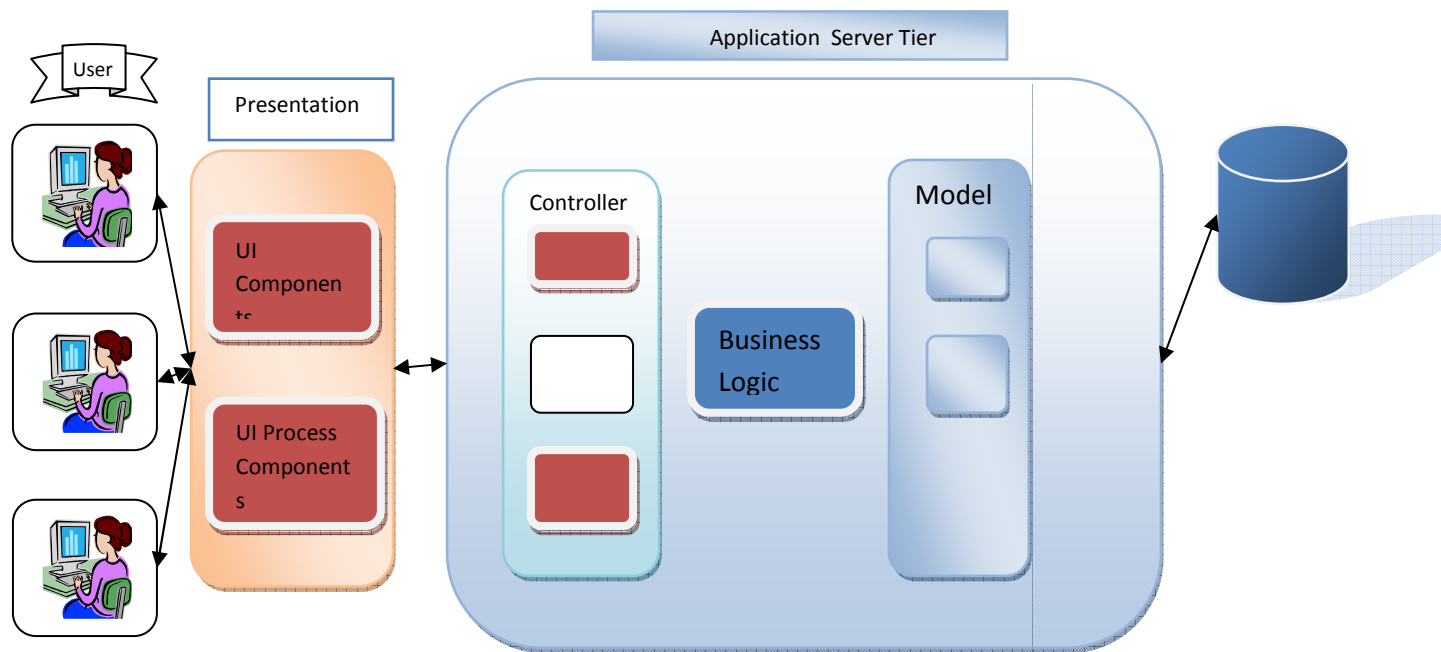
Figure: High level components view of STEM application

STEM application follows a ASP.NET MVC architecture style, ASP.NET MVC Framework includes a URL mapping component that enables building applications with clean URLs; The application defines controllers that contains number of predefined actions where each action process specific request, the process sequence includes executing application logic and retrieving data from the

CORPUS e-Ready Solutions (P) Ltd.
(The E-Governance Division of Corpus Software PLC)

domain model up to generating the response through a view component. The framework automatically maps URL's with friendly names ("/controller name/controller action/action parameters") to action in the controller class and invokes the action with the proper parameters
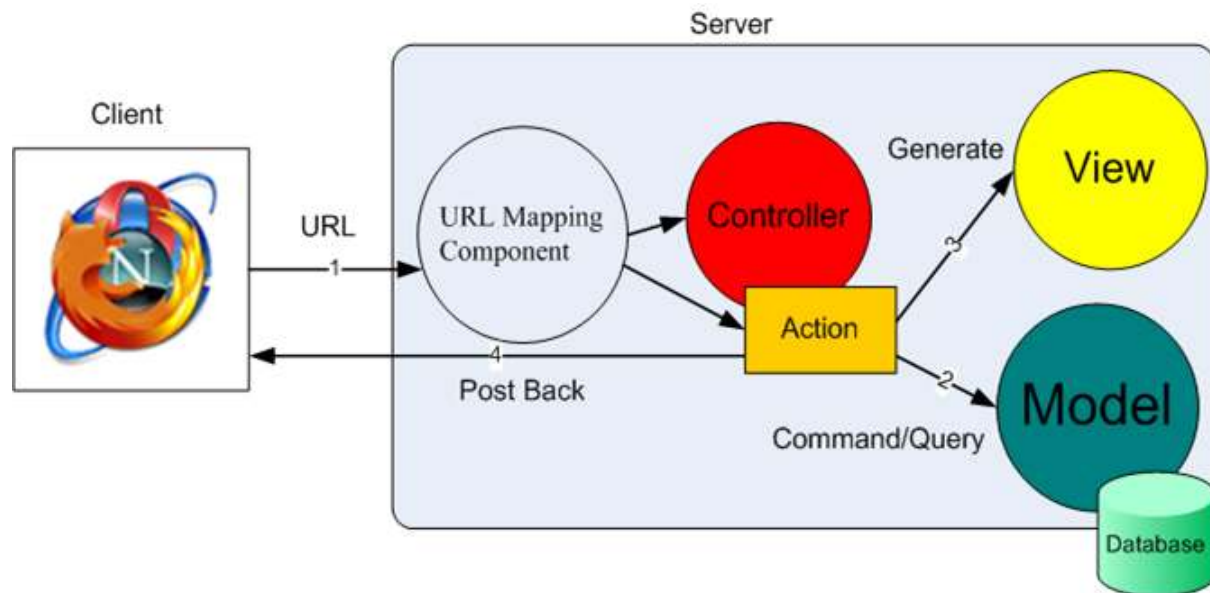
Asp.net MVC architecture style has following characteristics:

  - The ASP.NET MVC framework maps URLs to server code differently from a typical ASP.NET web site. Instead of mapping URLs to ASP.NET pages or handlers, the framework maps URLs to controller classes
  - Controller classes, then, handle incoming requests, such as user input and interactions, and execute appropriate application and data logic, based on user input. (ASP.NET MVC controllers implement a pattern known as the Front Controller pattern). A controller class typically calls a separate view component that generates HTML output as the response.
  - The ASP.NET MVC framework does not use the ASP.NET Post back model for interactions with the server. Instead, all end-user interactions are routed to a controller class. This maintains separation between UI logic and business logic and facilitates testability. As a result, the ASP.NET view state

and ASP.NET page life-cycle events are not integrated with MVC-based views.

Also, the MVC framework doesn't consider any URL as the endpoint to a physical server file to parse and compile to a class. In ASP.NET Web Forms, you have a 1:1 correspondence between a URL and a resource. The only exception to this rule is when you use completely custom HTTP handlers bound to a particular path.

3.1 Presentation Layer

Presentation layer components implement the functionality required to allow users to interact with the application. The following types of components are included in the presentation layer.

- User interface (UI) components: These components provide the mechanism for users to interact with the application. They format data and render it for display, and acquire and validate data entered by users.
  - UI process components: To help synchronize and orchestrate user interactions, it will be useful to drive the process using separate UI process components. This prevents the process flow and state management logic from being hard-coded into the UI elements themselves, and allows reusing the same basic user interaction patterns in other user interfaces.
  - UI processing components are not always necessary; create them only if you need to perform significant processing in the presentation layer that must be separated from the UI controls. Be careful not to mix business and display logic within the process components; they should be focused on organizing user interactions with your UI.

  A help icon will be provided on pages which include data entry such as new user registration, new agency registration, sanction generation etc. Clicking this icon will display the help content as defined by the super admin. Super admin can create a content specific to each page using site content functionality.

3.2 Controller

MVC controllers are responsible for responding to requests made against an ASP.NET MVC website. Each browser request is mapped to a particular controller. For example, imagine that you enter the following URL into the address bar of your browser:

http://localhost/Product/Index/3

For example, the controller might return a particular view back to the browser or the controller might redirect the user to another controller.

3.2.1 Understanding Controller Actions

A controller exposes controller actions. An action is a method on a controller that gets called when you enter a particular URL in your browser address bar. For example, imagine that you make a request for the following URL:

http://localhost/Product/Index/3

A controller action must be a public method of a controller class. C# methods, by default, are private methods. Realize that any public method that you add to a controller class is exposed as a controller action automatically (You must be careful about this since a controller action can be invoked by anyone in the universe simply by typing the right URL into a browser address bar).

There are some additional requirements that must be satisfied by a controller action. A method used as a controller action cannot be overloaded. Furthermore, a controller action cannot be a static method. Other than that, you can use just about any method as a controller action.

3.2.2 Understanding Action Results

A controller action returns something called an *action result*. An action result is what a controller action returns in response to a browser request.

The ASP.NET MVC framework supports several types of action results including:

1. ViewResult – Represents HTML and markup.
2. EmptyResult – Represents no result.
3. RedirectResult – Represents a redirection to a new URL.
4. JsonResult – Represents a JavaScript Object Notation result that can be used in an AJAX application.
5. JavaScriptResult – Represents a JavaScript script.
6. ContentResult – Represents a text result.
7. FileContentResult – Represents a downloadable file (with the binary content).
8. FilePathResult – Represents a downloadable file (with a path).
9. FileStreamResult – Represents a downloadable file (with a file stream).

All of these action results inherit from the base ActionResult class.

In most cases, a controller action returns a ViewResult. For example, the Index controller action in Listing 2 returns a ViewResult

3.3 Model

An MVC model contains all of the application logic that is not contained in an MVC view or MVC controller. In particular, an MVC model contains all of your application business and data access logic.
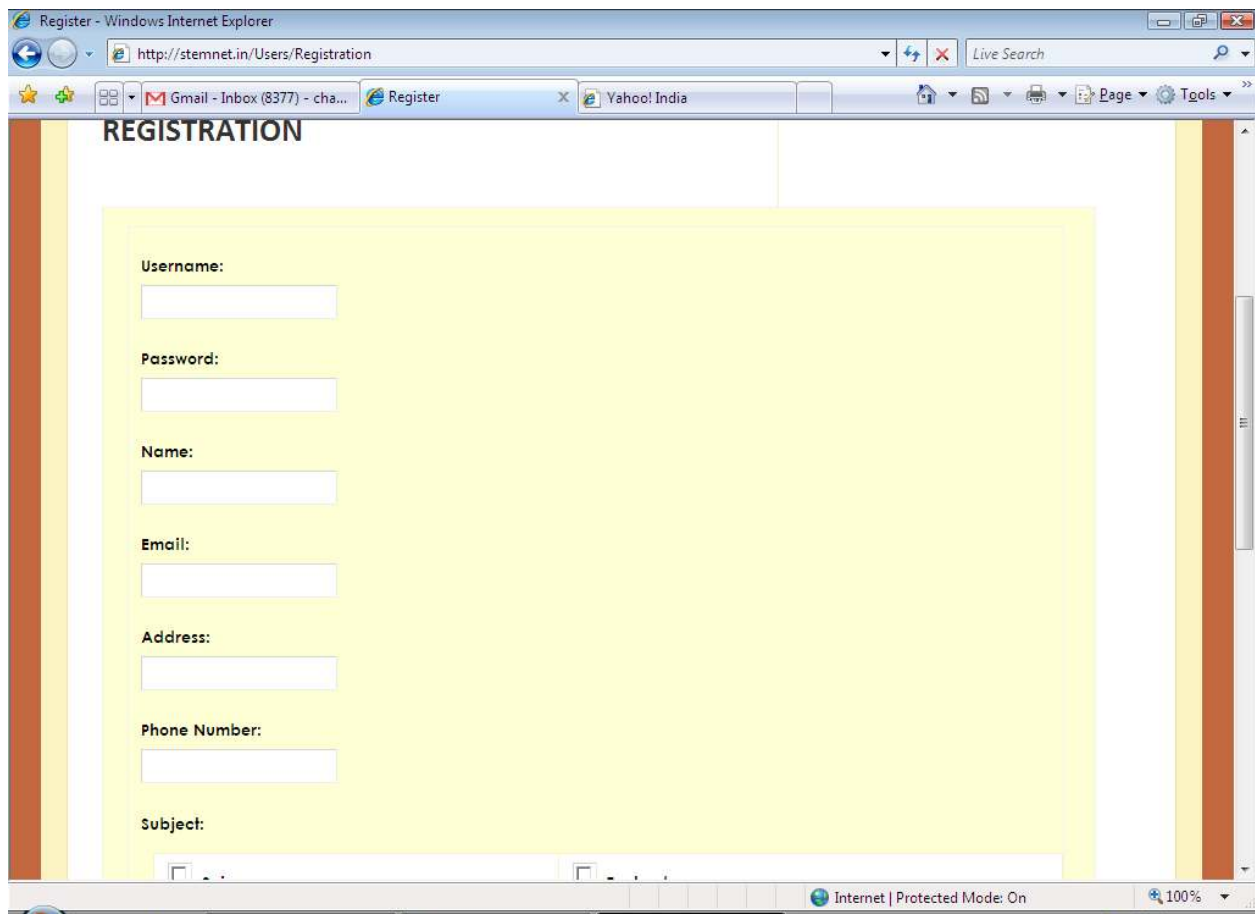
You can use a variety of different technologies to implement your data access logic. For example, you can build your data access classes using the Microsoft Entity Framework, NHibernate, Subsonic, or ADO.NET classes.

In this, we use LINQ to SQL to query and update the database. LINQ to SQL provides you with a very easy method of interacting with a Microsoft SQL Server database. However, it is important to understand that the ASP.NET MVC framework is not tied to LINQ to SQL in any way. ASP.NET MVC is compatible with any data access technology.

4. Student Corner

This module contains the functionality related to Student Activity in the system. This includes new Student registration, e-Class, Content Reading, Chat, Friend Request,-Control Panel, Blog, KB.

4.1 Student Registration

CORPUS e-Ready Solutions (P) Ltd.
(The E-Governance Division of Corpus Software PLC)

This Page is for student Registration. After registration student get a verification mail form
info@stemnet.in. Student has to verify the mail id.

Following table lists the elements that would be part of student registration form.

Field Name	Input Type	Input Validation	Description/Comments
User Name	Text Box	Mandatory	This is user login id . two User id not be same
Password	Text Box	Mandatory	This is the User Password
Name	Text Box	Mandatory	This is Name of User Complete Name
Email	Text Box	Mandatory	A Valid email Id of user. A verification mail is send to this Id and this id is used as future communication.
Address	Text box	Mandatory	This is communication Address of the User.
Phone No.	Text box	Mandatory	Phone No of User.
Subject	Check Box	At least one check box is checked	Subject for which user want to registration.
Class	Dropdown	Mandatory	Class of the Student
School	Text Box	Mandatory	School Name
Board	Text box	Mandatory	Name of the Board
Create	Button	Not Applicable	Clicking this button will validate the inputted data on the form using client side java script

			functions. If the data is valid then it is posted to server for checking whether a Username is already exists in the system. If Not Then it stores The user information and display a Waiting Message for Email Validation.
Reset	Button	Not applicable	Reset The Date of this Page.

Business Logic for Registration

1. Username field Must Be Check in Database. Two users can't Have Same Username. For this we have to check in User Registration Table.
2. A Verification Mail must be send to User give Mail Id.
3. By default Activation of users is False. Only Admin can active the users.

4.2 Student Login

This Page is For Login OF Student:
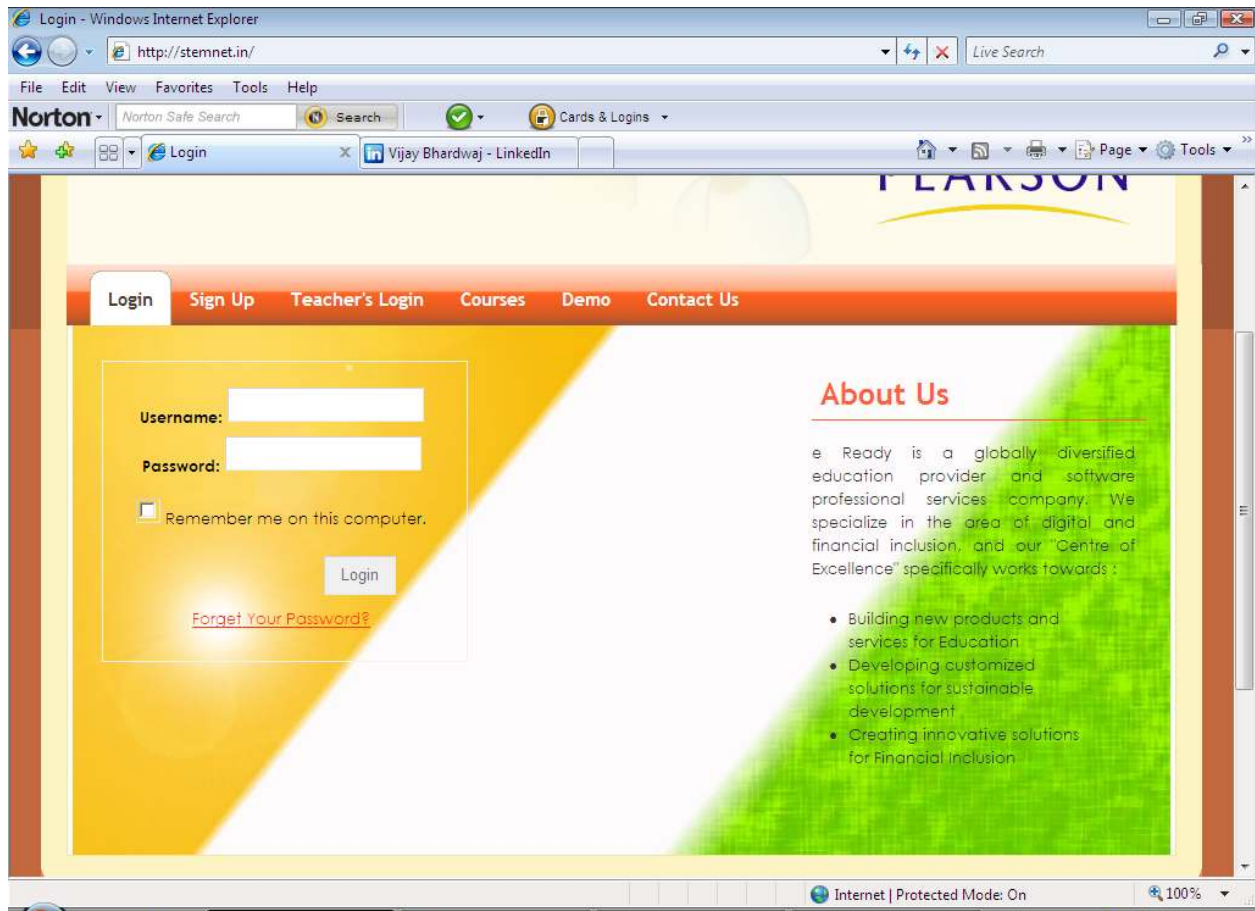
Figure 1 : Login Page

Following table lists the elements that would be part of student Login form

Field Name	Input Type	Validation	Description/Comments
Username	Text Box	Mandatory	This field Takes Users login ID

Password	Text Box	Mandatory	This Field Takes Users Password.
Remember Me	Check Box	Not Applicable	This is used for Create cookies For save Password
Forgot Password	Hyper link	Not applicable	Clicking on Link redirect user to user forget Password Page.
Login	Button	Not applicable	Clicking this button will validate the inputted data on the form using client side java script functions. If the data is valid then it is posted to server for checking whether a Username and password is Correct. If Correct then Redirect User to Control Panel Page else Redirect to Login Page with error Message.

Business Logic for Login

1. Check User Name and Password Must be same as In Database.
2. Redirect user to Forget Password Page when User Click on Forget Password link.

CORPUS e-Ready Solutions (P) Ltd.
(The E-Governance Division of Corpus Software PLC)

4.3 Control Panel

This Is the Home Of this Application. We Name it 'Control Panel'.

On this Page student Have:

- Performance Dial.
- Calibration Dial.
- Button for Science, Math, English, Technology.
- News Feed Form NASA.
- Resource Panel.
    - e Class
    - Chat
    - Student's Corner
    - Teacher's Corner
    - Knowledge Bank (KB)
    - Progress Tracker
    - Connect to Mentor
- Test.
- Inbox (for Stem net Mail)
- Profile
    - Group List
    - View Profile
    - Buddy Zone
    - Buddy Request Status.
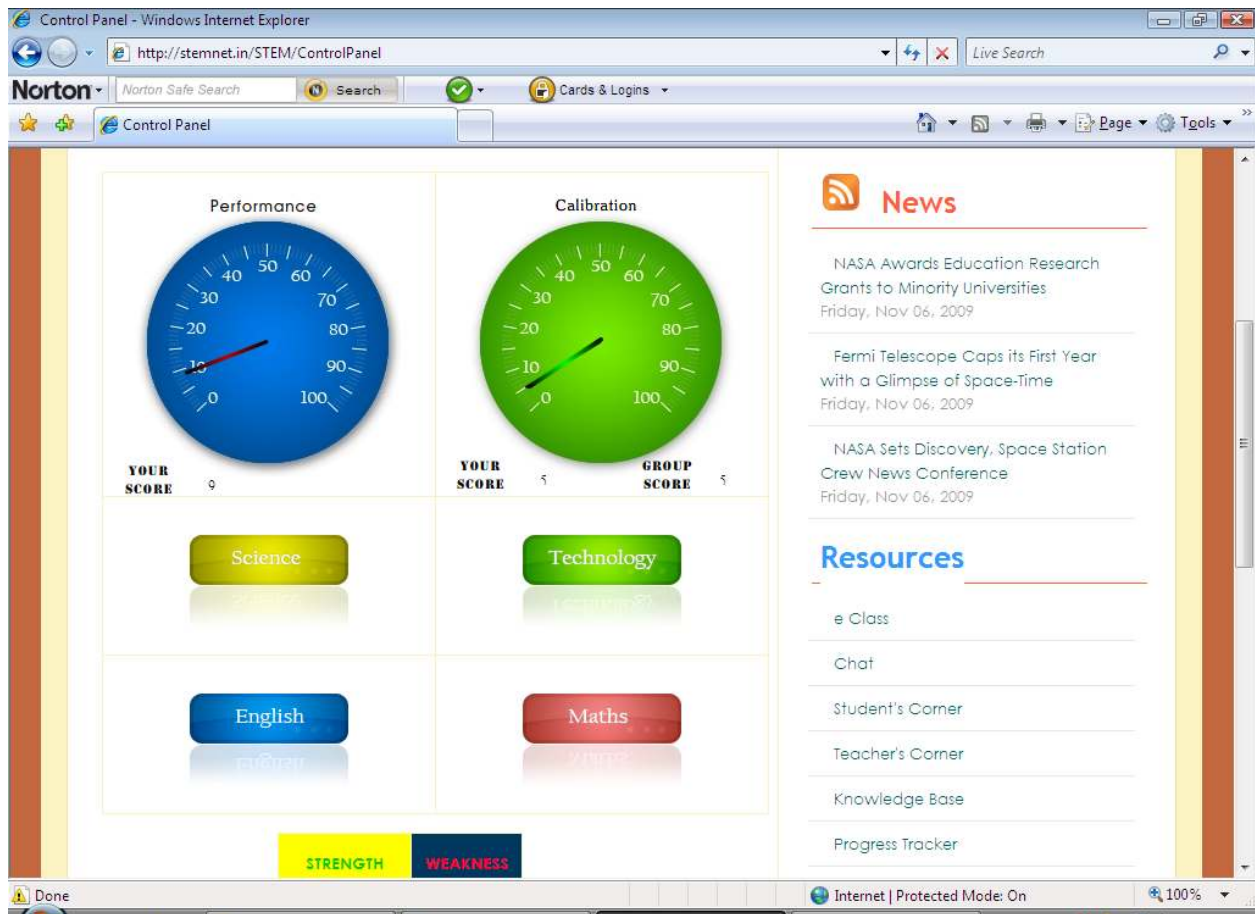
4.3.1 Performance Dial

The Performance Dial show the student last Score in Test.   This is flash based Dial. To show this we need Flash player.

4.3.2 Calibration Dial

Calibration Dial Show student average Performance with all student. Means Performance of Student in Class Room. This is flash Base Dial.

4.3.3 Button for Science, Math, English, Technology.

These Buttons are for Content reading.

- Science Button: This Button Redirect Student to Select Science: Chemistry and physics.
- Math Button: This button Open Current Uploaded Math Content.
- English Button: Open English Content.
- Technology Button: Open technology Content.

4.3.4 News Feed Form NASA

This Section shows RSS feed Of NASA.

4.3.5 Resource Panel

4.3.5.1 e-Class

e Class is the service, which uses the mechanism of e learning but blended in an online, real time coaching by subject specialists on a 1-1 or 1-N with a student via common a technological platform.

This service is providing by "Dim Dim".

Feature of e class:

1. Virtual Class Room.
2. White Board
3. Public Chat and private Chat
4. Video and Audio Live Class room
5. All Controls are Controlled By teacher.
6. Desktop Share
7. File share (i.e. PPT , Flash file)

4.3.5.2 Chat

This is Public Chat Room. Each chat Text must Be store in Database so that Admin can Monitor Chat. List of Online student is Visible on Right side of the Chat. On Clicking On user Name U can see The User Profile.
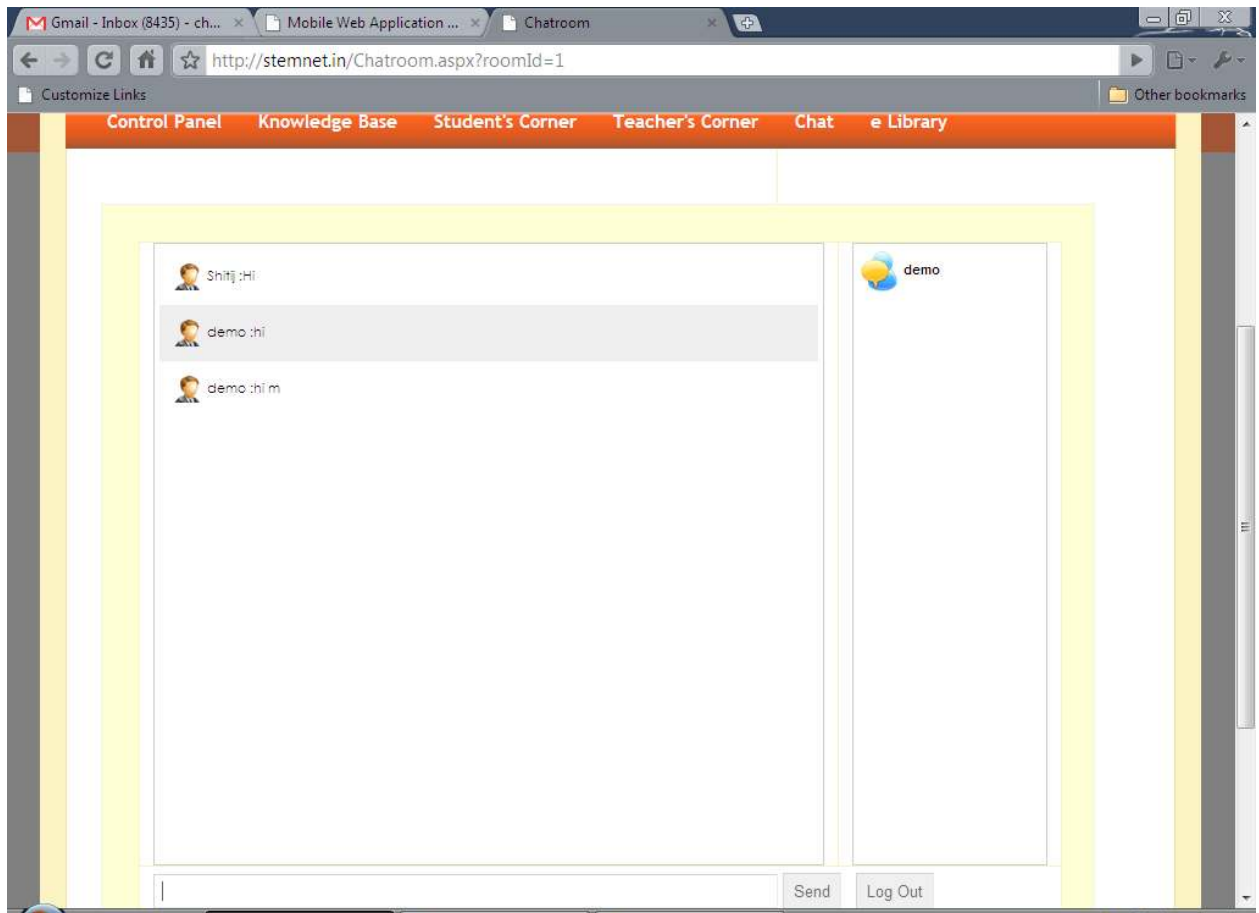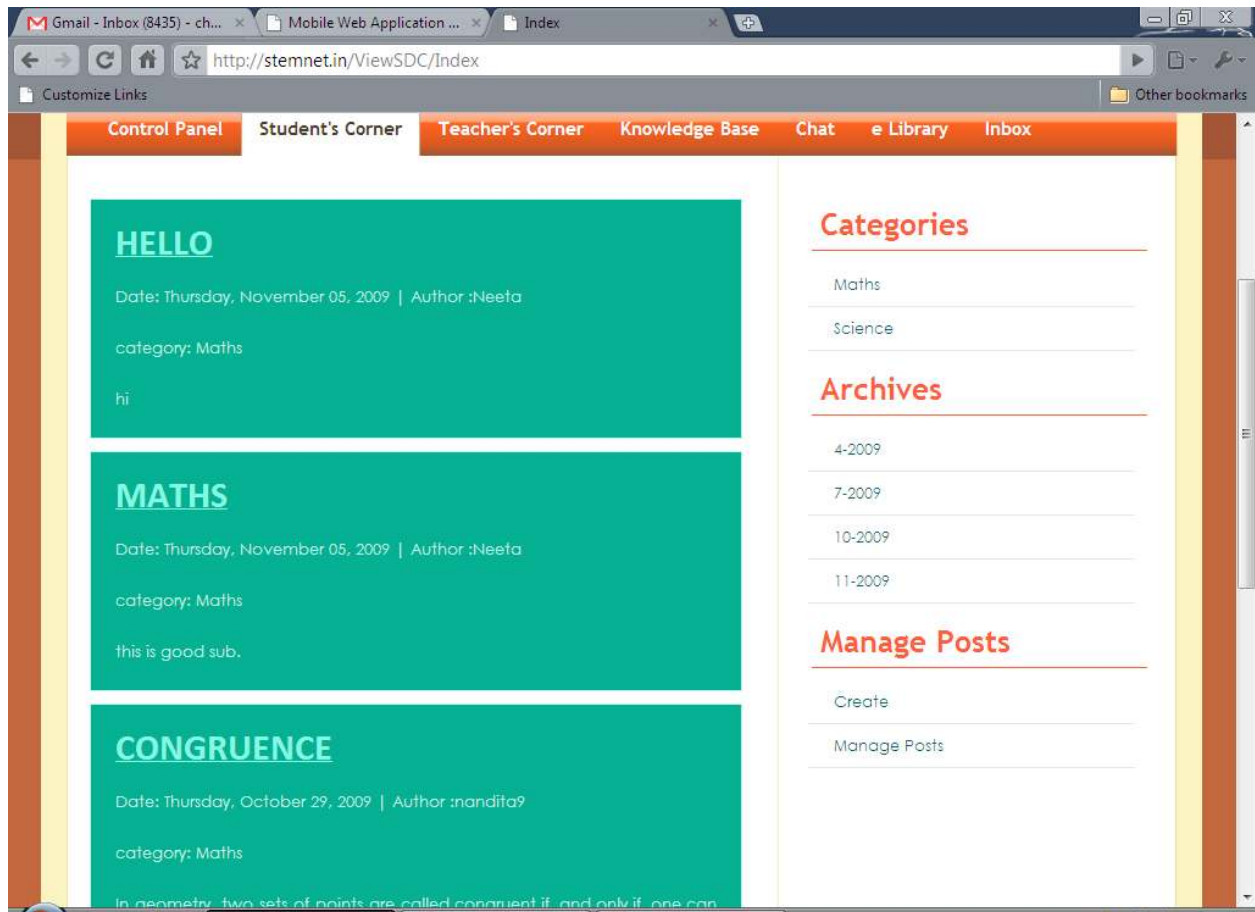
4.3.5.3 Student's Corner

This Is Blog of Student.   Student Can Create and See All Post . this Blog Content The Question's Of Student .
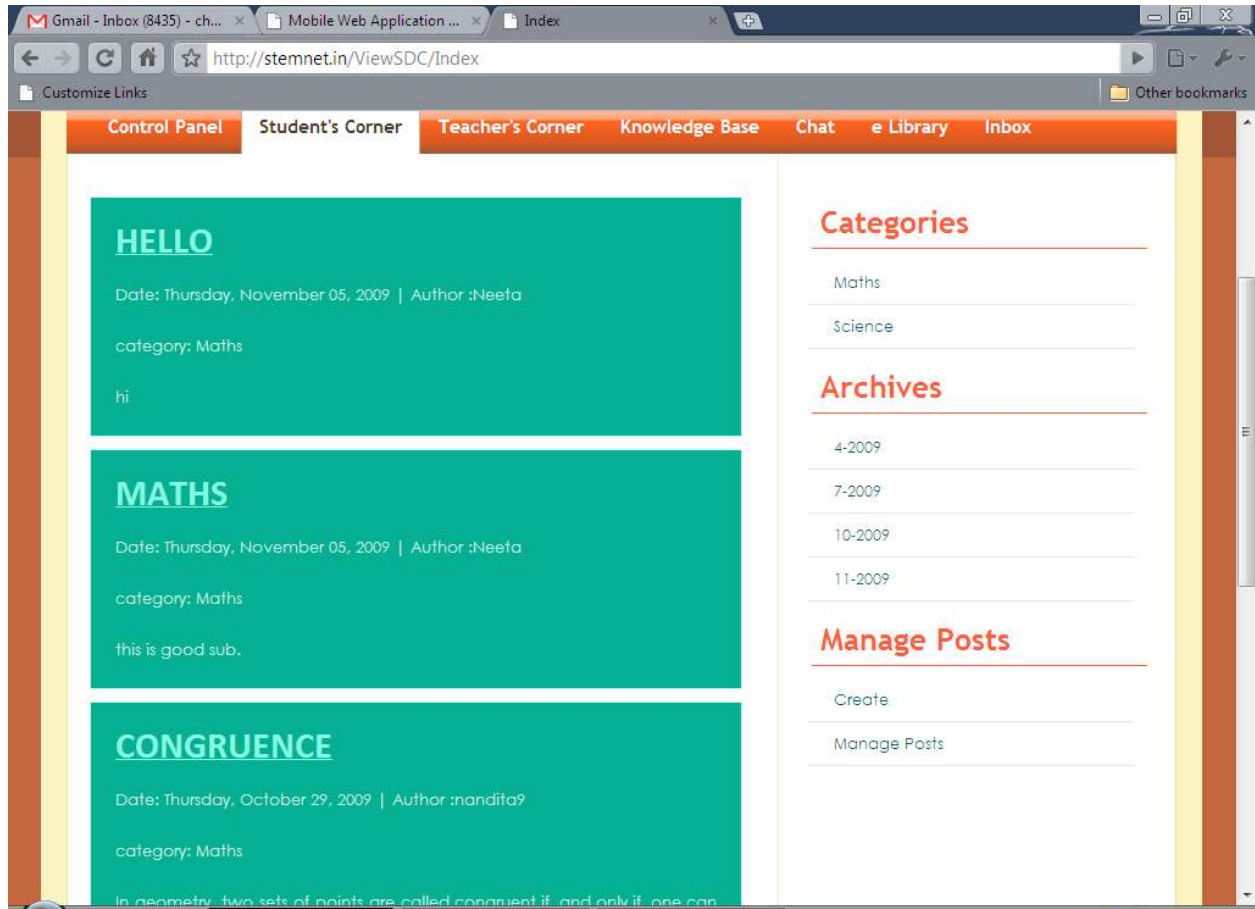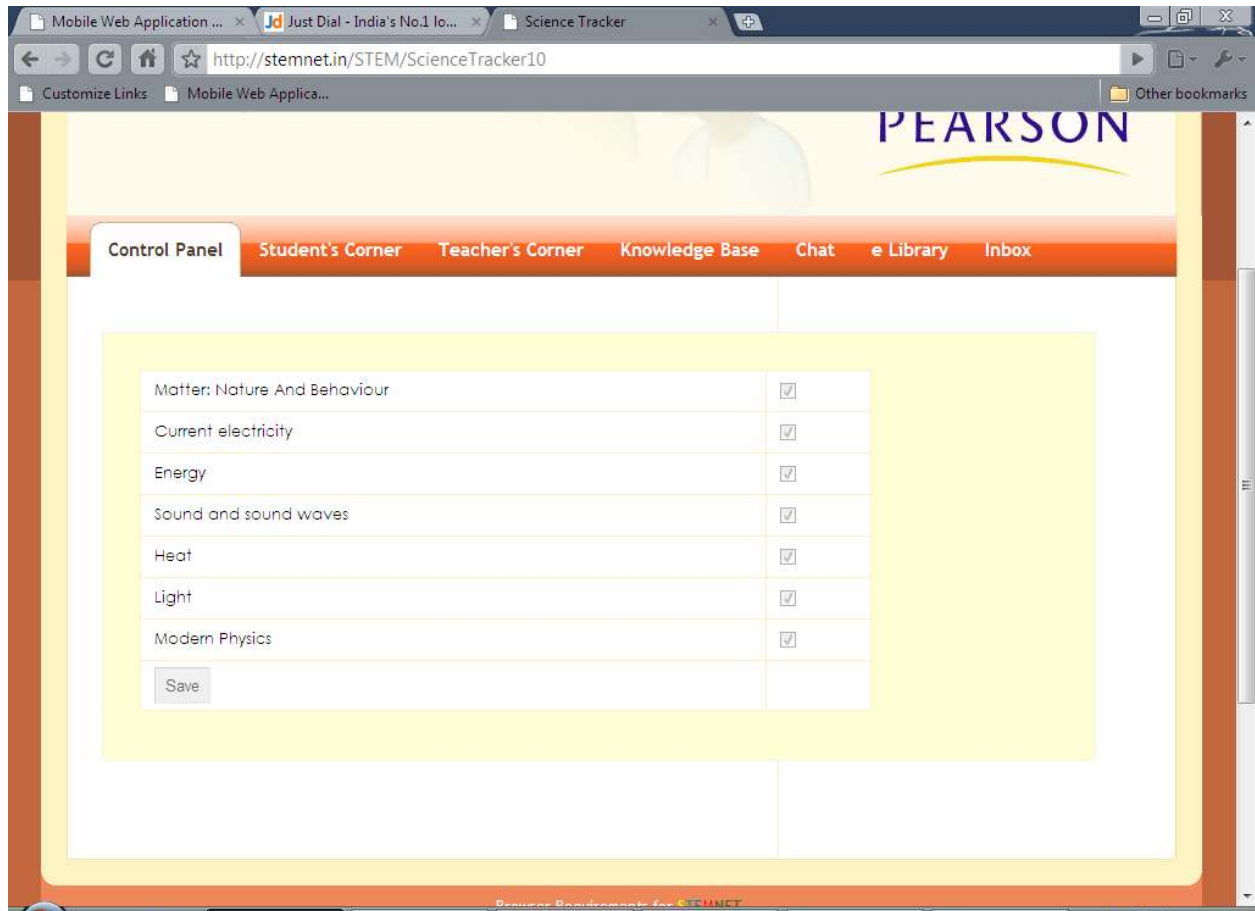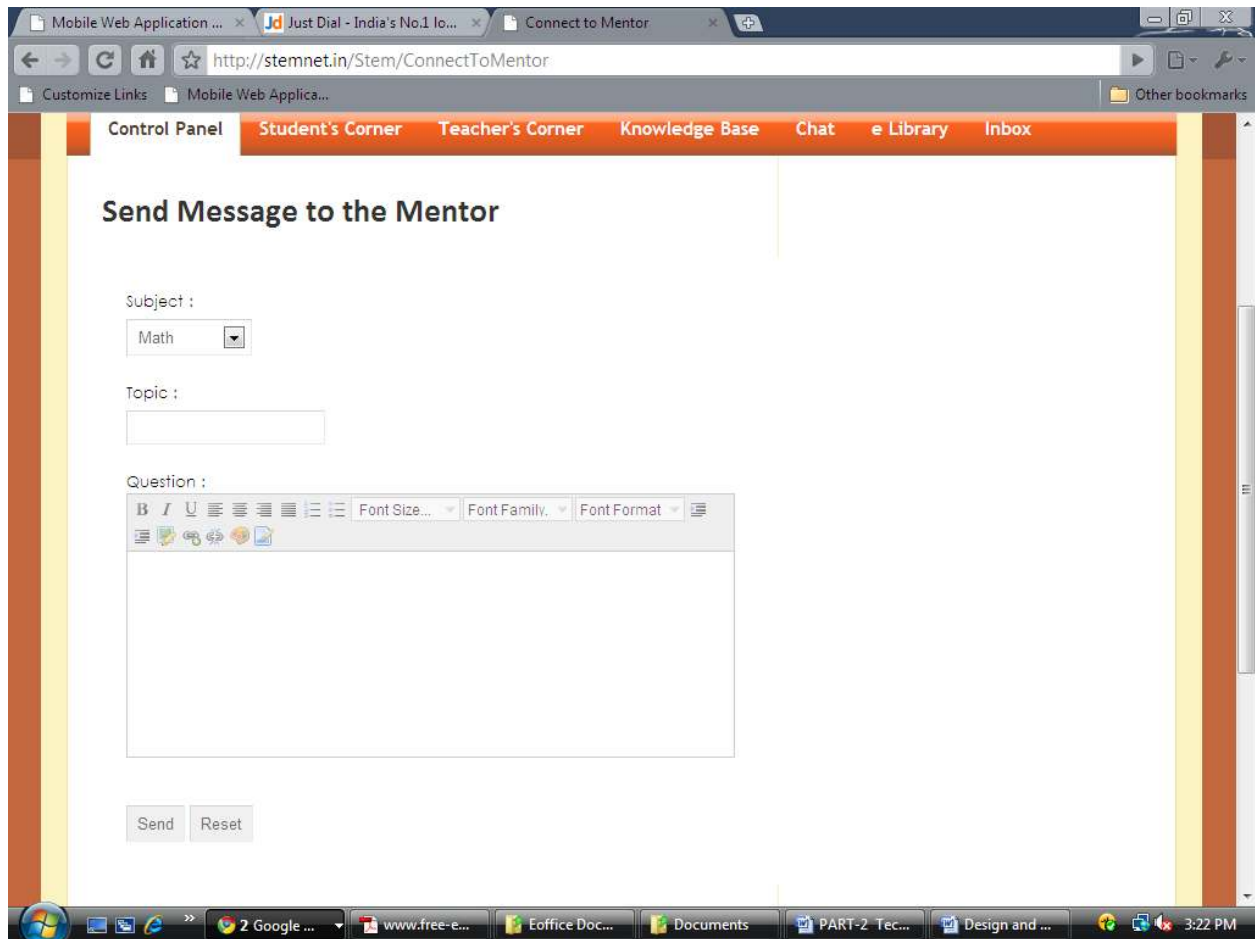

4.3.5.4 Teacher's Corner

This Is Blog of Teacher's.   Student Can Create and See All Post. This Blog Content the Question's Of Student.

4.3.5.5 Progress Tracker :

4.3.5.6 Connect Mentor:

4.3.5.6 Friend Request