**1. Introduction**

* **Project Profile:**

|  |  |  |
| --- | --- | --- |
| **Project Title** | **:** COMPLAIN MANAGEMENT SYSTEM | |
| **Platform** | **:** Microsoft Visual Studio 2012, MS SQL Server 2012 | |
| **Organization** | **:** Shree Ramkrishna Institute of Computer Education | |
|  |  | and Applied Science |
| **Developed By** | **:** | Nainesh Thummar |
| **Project Guide** | **:** | Mr. Jayesh Pushtiwala |
| **Front End** | **:** ASP.net C#(Visual studio 2012) | |
| **Back End** | **:** Microsoft SQL Server Management Studio 2012 | |
| **Technology** | **:** Master Page with C#,BOOTSTRAP and CSS | |

**2. System Introduction**

* **System Definition**
* The main purpose of “TASK MANAGEMENT SYSTEM” is to manage the allotment of task to the Faculty given by HOD (admin). Admin will keep the track of the update of tasks updated by the faculty of respective task.
* Also the Status of task can be known by the admin.
* The entire faculty can see their particular task and also give their task working updates or status.
* Their login to the system will be according to their designation. Accordingly, they can access the information.
* Every task assigned to faculty has particular TASK\_ID through which every record and update can be tracked and kept detail of it. Assignment of Task to particular faculty will also have the ASSIGN\_FACULTY\_ID of faculty so as to match the track of faculty to whom task is assigned.
* Task will have detail of starting and ending date of TASK as decided by the Admin.
* Any Faculty or user can upload their specific work to server in formats like pdf, doc, xls, jpeg, etc.
* Admin can download the files, uploaded by a specific user and can view their status.
* Admin can also monitor the details of task assigned to users, task load, etc using graphs and pie charts.
* Admin can generate a report in many suitable formats (pdf, doc, xls, etc) to keep records.
* **System Objective**
* In this system admin will assign the TASK to the appointed faculty under his authority. Admin will have full right to create the task, assign to faculty, check the status of task and download a file from server.
* Whereas faculty can give update of task and its status. When admin assign a task, a mail will send to the particular faculty.{mail notification}
* Faculty can register for their account.
* It is simple, manageable SYSTEM for TASK MANAGEMENT which helps to release the pressure of managing it manually.
* Moreover, Graphs and Pie Charts can help admin to monitor the work load and tasks assigned to faculties or users, so that every user gets equal work and overload on only one faculty can be avoided.
* Also, faculty or user can upload their work in many formats (pdf, xls, doc, etc.), which makes easy to share files with the admin.
* JavaScript is used to disable “Back Button”, so user can’t login back, after logout, without passing though the login page and also sessions are used on required pages so, this makes site secure.
* Admin can check the all the task details like which are not assigned, which are assigned and which are completed, and also filter the record of task as per main task, faculty, status.
* Finally, admin can generate a filtered report of the Tasks Assignment, and download it in formats like pdf, doc and xls, which helps to keep daily, monthly or yearly records, as per needed.
* **System Scope**

* WORK ASSIGNMENT AND STATUS TRACKING SYSTEM is developed for college for assigning and maintains task and their respective records.
* The system uses any places that are divides with department, like any company where work divides with department we can manage particular department task using this software.
* The system also keeps both sides information about status of their task.
* Graphs used in it, can reduce the efforts of monitoring huge data, which is found in big companies.
* Daily, Monthly or yearly reports can be generated, which helps in keeping record of work in a company.
* The security of the system is maintained due to sessions, so it can be expandable to some high administrative places.
* **Hardware & Software Requirements (with justification)**
* **Hardware Requirement**

1. **Hard Disk:** 500GB
2. **RAM:** 4GB

* **Software Requirement**

1. **Front End:** ASP.net C# (Visual studio 2012)
2. **Back End:** Microsoft SQL Server Management Studio 2012

* Asp.net C# provides better performance by taking advantage of early binding, just-in-time compilation, native optimization, and caching services right out of the box.
* Web application exists in compiled form on the server so the execution speed is faster as compared to the interpreted scripts.
* SQL LocalDB can be used as a developer environment for applications targeting all editions of SQL server.
* SQL LocalDB enables features such as stored procedures, user-defined functions and aggregates, .NET Framework integration, spatial types and others that are not available in SQL server compact.

**3. Requirement Analysis& Modeling**

* **Expected Working of system**
* TASK MANAGEMENT SYSTEM is used for assigning and maintaining task and their respective records.
* H.O.D. adds work, then assigns work and system will notify to faculty via email.
* All faculty have their own username password, they can see their task and then update status.
* They can see their all particular task details.
* Charts and Graphs work in real time as per the database.
* HOD can see all status of all faculties.
* H.O.D. can download reports and specific files uploaded by other users from server.
* Similarly, work related files can be uploaded to server by faculties or users.
* **Data flow Diagram**

**0 level DFD**

Add and assign task

HOD

Give status of task,

Upload file for task

FACULTY

**1st level DFD**

Receive Task,

View Own task Detail

View Task Detail,

Status details of all task,

File detail of all task, Graph of task detail

Add task

Store detail

Retrieve detail

TASK DETAIL

View task

Task detail

Retrive

FACULTY DETAIL

Assign to

faculty

Update

Task

Detail

HOD

View task detail

Data of Faculty

FACULTY

Assign

Task

Receive

task

Can see personal status

Give file ,Status

HOD can view all detail

Take

Task

detail

Store file detail

Retrieve Detail

Retrieve file detail

TASK STATUS DETAILS

Store details

FILE DETAILS

HOD can view task in graph format and download reports

Retrieve information for graph and reports

**2nd level DFD OF PROCESS 3**

Update task details

TASK DETAIL

Retrieve task detail

Can see personal task details

HOD can view all Status detail of any Project

Give Status

Store status detail

FACULTY

Retrieve details

HOD

TASK STATUS DETAIL

Can View file detail and download file

Upload file For task

Retrieve file detail

Retrieve Task detail

Store file details

FILE DETAILS

**2nd level DFD OF PROCESS 4**

TASK DETAIL

Retrieve task detail

Can view and download report

Retrieve faculty detail

FACULTY DETAIL

HOD

View detail in graphical form

Retrieve task detail for calculations

* **Process Specification**

# Process 1:

|  |  |
| --- | --- |
| **NAME:** | Task management |
| **TYPE:** | Online |
| **DESCRIPTION**: | It will take the input of task detail  Like, task name and its description from HOD  And save the data on task detail data store  Also HOD can see the detail of all task details from  Task detail, that comes from the task detail data store. |

# Process 2:

|  |  |
| --- | --- |
| **NAME:** | Assign task |
| **TYPE:** | Online |
| **DESCRIPTION**: | HOD can assign the task to the faculty,  Faculty data comes from faculty data store,  And task that are not assign comes from task detail,  Faculty receive particular task  And the task assign to faculty that store in task detail data store, |

# Process 3.1:

|  |  |
| --- | --- |
| **NAME:** | Status Management |
| **TYPE:** | Online |
| **DESCRIPTION**: | Faculty give status for particular task ,  And that store in status data store,  Also it will update the status of task from task detail data store,  All the status that will given by faculty can see by HOD  HOD can see all the status update, only can see. |

# Process 3.2:

|  |  |
| --- | --- |
| **NAME:** | File Management |
| **TYPE:** | Online |
| **DESCRIPTION**: | Faculty upload file for particular task ,  And that store in a particular folder,  also detail of file store in file detail data store,  All the file that will upload by faculty can see by HOD  HOD can see all the detail of file and download it. |

# Process 4.1:

|  |  |
| --- | --- |
| **NAME:** | Report Management |
| **TYPE:** | Batch |
| **DESCRIPTION**: | HOD can view report of all task categorizes wise and also download a report in form of PDF , EXCEL and WORD, here data comes from Task detail data store and faculty data store,  Here input is given by system from task detail table. |

# Process 4.2:

|  |  |
| --- | --- |
| **NAME:** | Graph Management |
| **TYPE:** | Batch |
| **DESCRIPTION**: | HOD can view graph in form of pie chart and column chart , can view task detail and faculty assign task detail in a form of graphical view,  Here data calculation done by system and data comes given data stores. |

* **Data Dictionary:**

|  |  |
| --- | --- |
| **Name:** | Login table |
| **Alias:** | Faculty/HOD detail |
| **Where used/How used:** | Data use for particular login  Data use for task assign to faculty |
| **Content Description:** | Faculty\_login =f\_ id + name + contact + email + department + username + password + designation |

|  |  |
| --- | --- |
| **Name:** | List\_of\_work |
| **Alias:** | Task status detail |
| **Where used/How used:** | New task add(input)  When Task assign to faculty - update task detail  When Status given by faculty – update task detail |
| **Content Description:** | List\_of\_work = l\_id + task\_name + description + task\_status + assign\_f\_id + assign\_date + last\_date + progress + submitted\_date |

|  |  |
| --- | --- |
| **Name:** | status\_of\_work |
| **Alias:** | task detail |
| **Where used/How used:** | Status update (input)  Faculty can see its personal status- status management(output)  HOD can see all status-status management(output) |
| **Content Description:** | Status\_of\_work = s\_id + l\_id + status\_date + status\_description + work\_progress |

|  |  |
| --- | --- |
| **Name:** | File\_detail |
| **Alias:** | File details |
| **Where used/How used:** | File upload(input)  Faculty upload file  File View (output)  HOD can see all file detail and download file |
| **Content Description:** | File\_detail = file\_id + file\_name + file\_ext + file\_path + upload\_date |

**4. Design**

* **Database design:**

Table: work

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Data type** | **Key** |
| work\_id | Int | Primary key |
| work\_type | nvarchar(50) |  |

Table: faculty\_login

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Data type** | **Key** |
| Faculty\_id | Int | Primary key |
| Name | nvarchar(50) |  |
| Contactno | nvarchar(50) |  |
| Email\_add | nvarchar(50) |  |
| Department | nvarchar(50) |  |
| Uname | nvarchar(50) |  |
| Passwrd | nvarchar(50) |  |
| Designation | nvarchar(50) |  |

Table: status\_of\_work

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Data type** | **Key** |
| Status\_id | Int | Primary key |
| List\_id | Int | Foreign key |
| Status\_date | Datetime |  |
| Status\_description | nvarchar(200) |  |
| Work\_progress | nvarchar(50) |  |

Table: list\_of\_work

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Data type** | **Key** |
| list\_id | Int | Primary key |
| Work\_id | Int | Foreign key |
| Task\_name | nvarchar(50) |  |
| Description | nvarchar(500) |  |
| Task\_status | nvarchar(50) |  |
| Assign\_faculty\_id | Int | Foreign key |
| Assign\_date | Date |  |
| Last\_date | Date |  |
| Progress | nvarchar(50) |  |
| Submission\_date | Date |  |

Table: file\_detail

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Data type** | **Key** |
| file\_id | Int | Primary key |
| file\_name | nvarchar(50) |  |
| file\_ext | nvarchar(50) |  |
| file\_path | nvarchar(200) |  |
| list\_id | Int | Foreign key |
| upload\_date | Datetime |  |

**ER Diagram(System Architecture):**

work

add

List\_o\_work

View/

add

update

faculty

Add/

view

Add/

View

Status\_of\_work

faculty

* **System Flowchart:**

Admin Side:

login

success

login

admin interface

fail

View task load chart

View Reports

View status detail of all

Assign task

Add main task

View File details

View task load chart

View all task detail

Add sub task and detail

manage

User Side:

login

success

login

View own task detail

user interface

fail

Update status

View completed task

View Recieve task

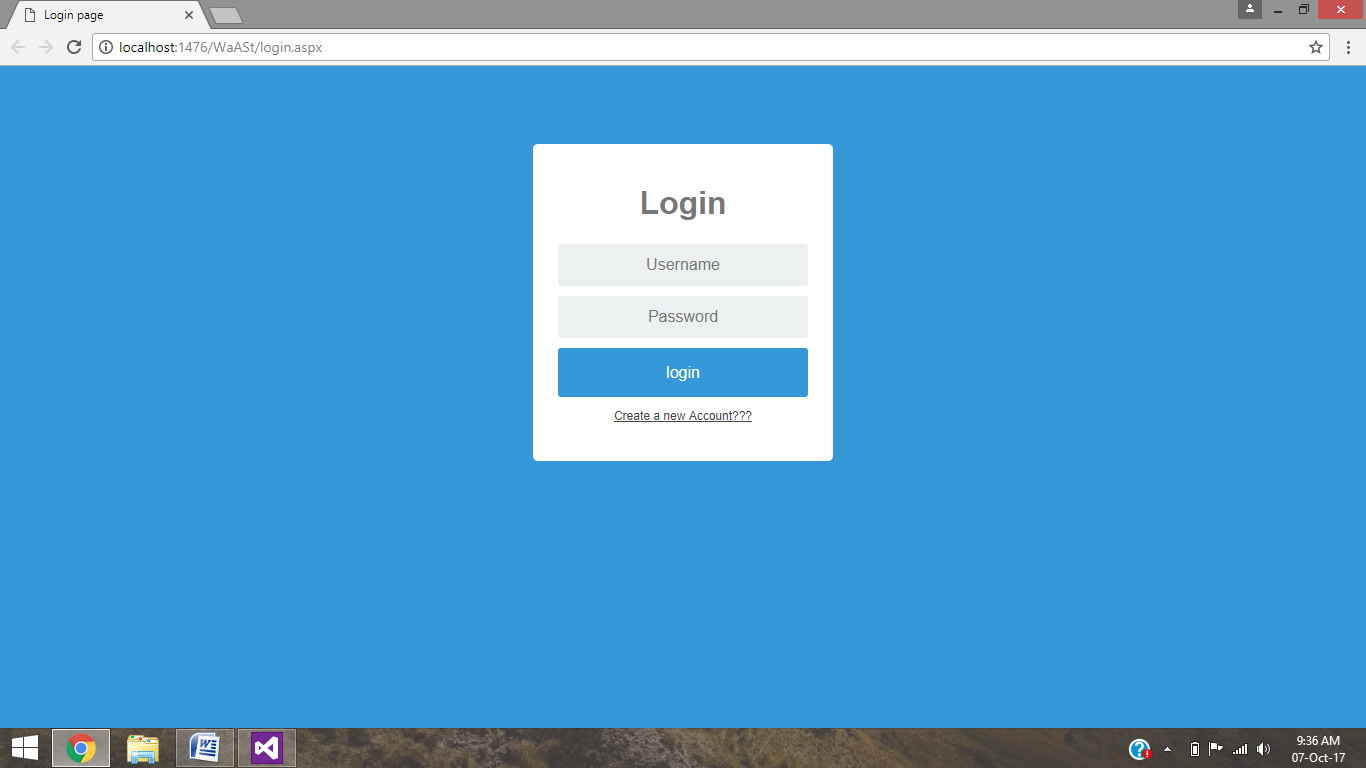
View Working task

File Upload

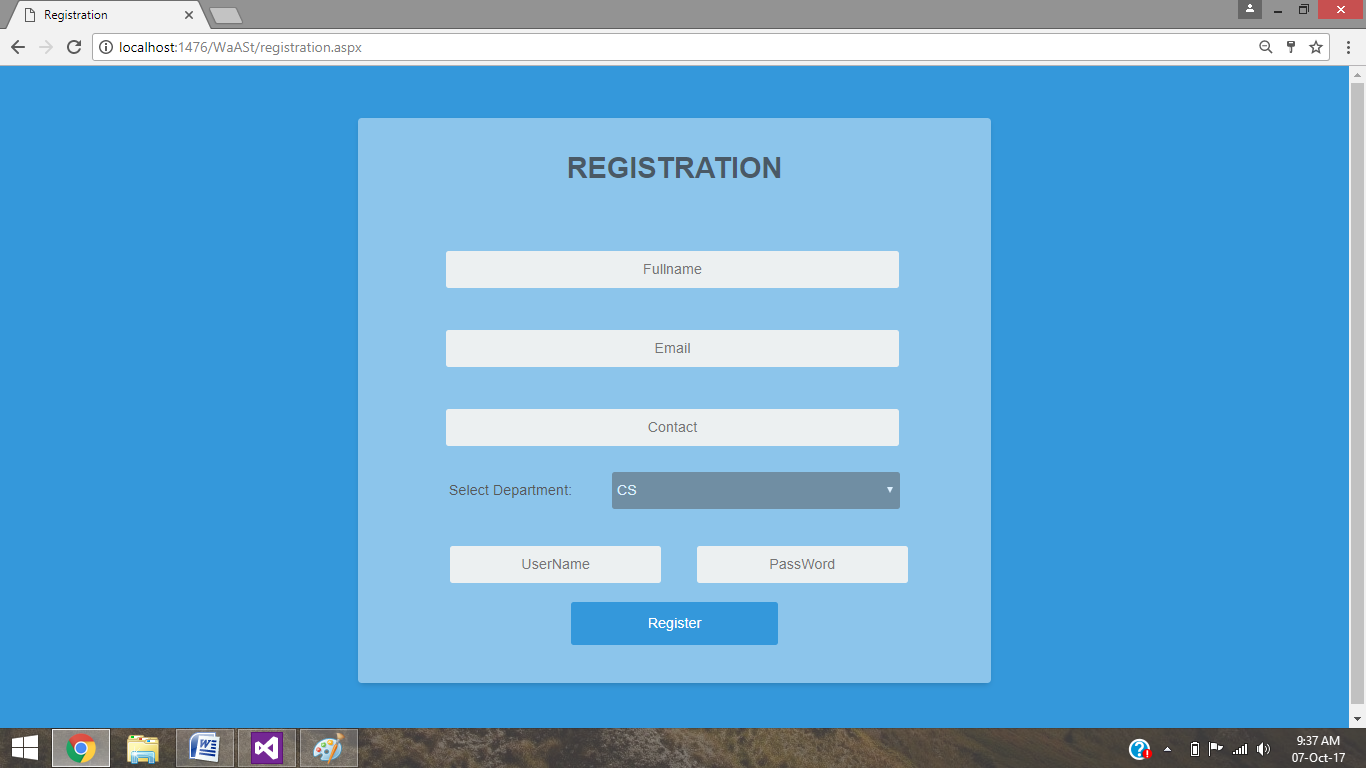
manage

* **Form/Report Design(Layout):**

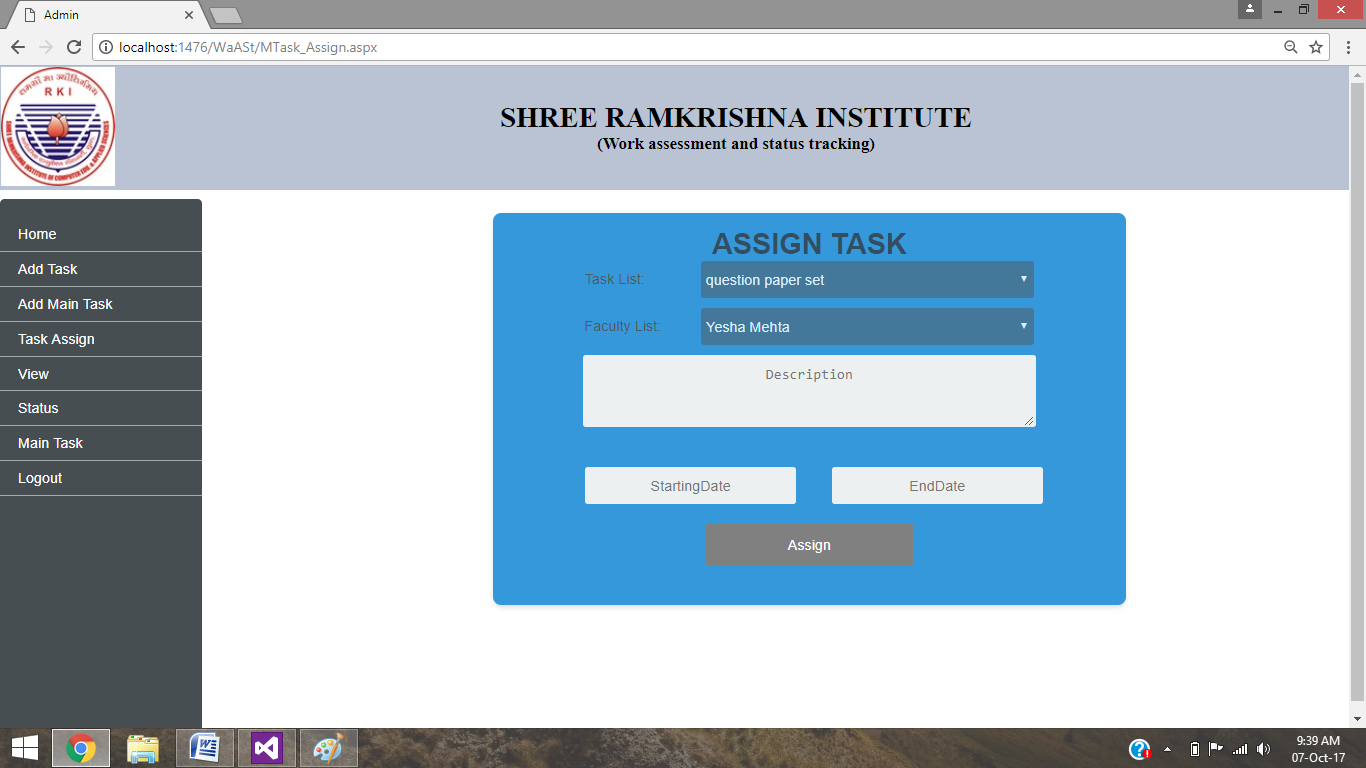
**Login Page:**

****

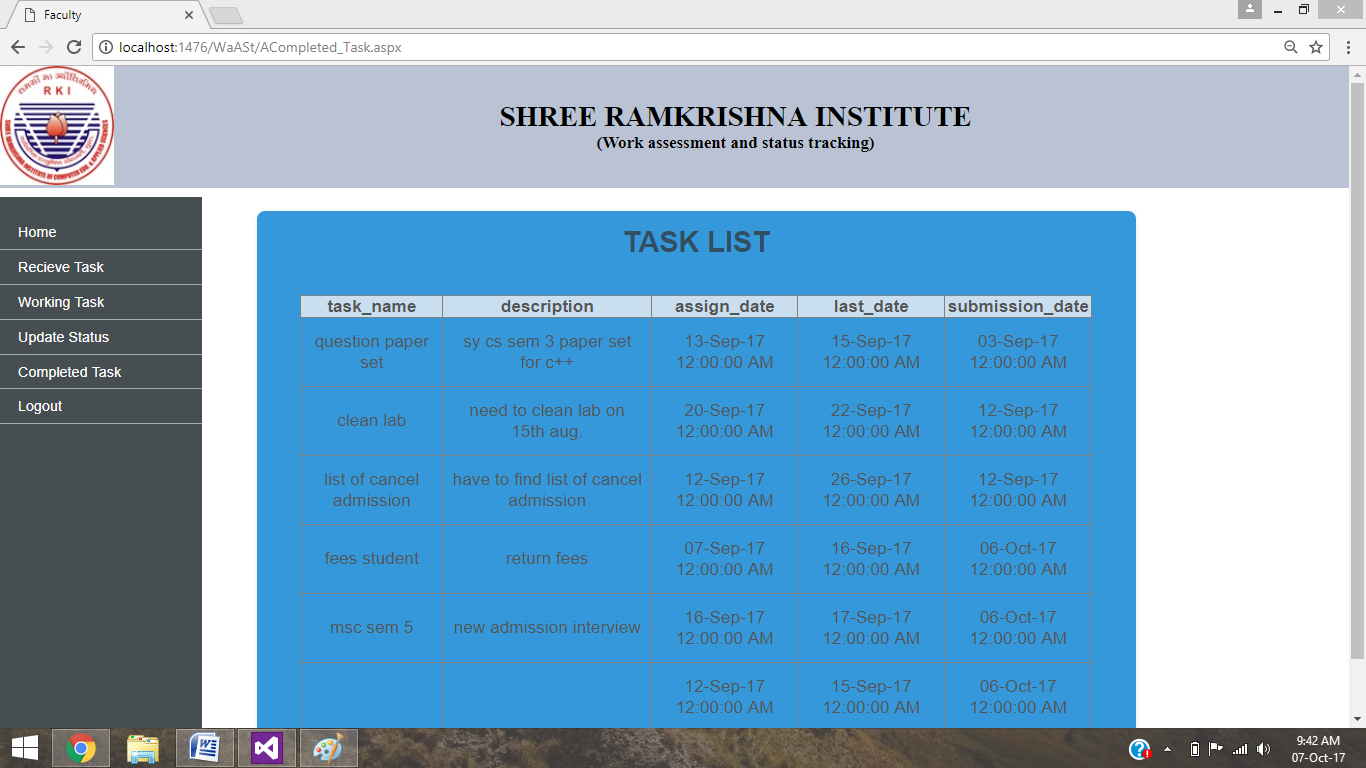
**Registration Page:**

****

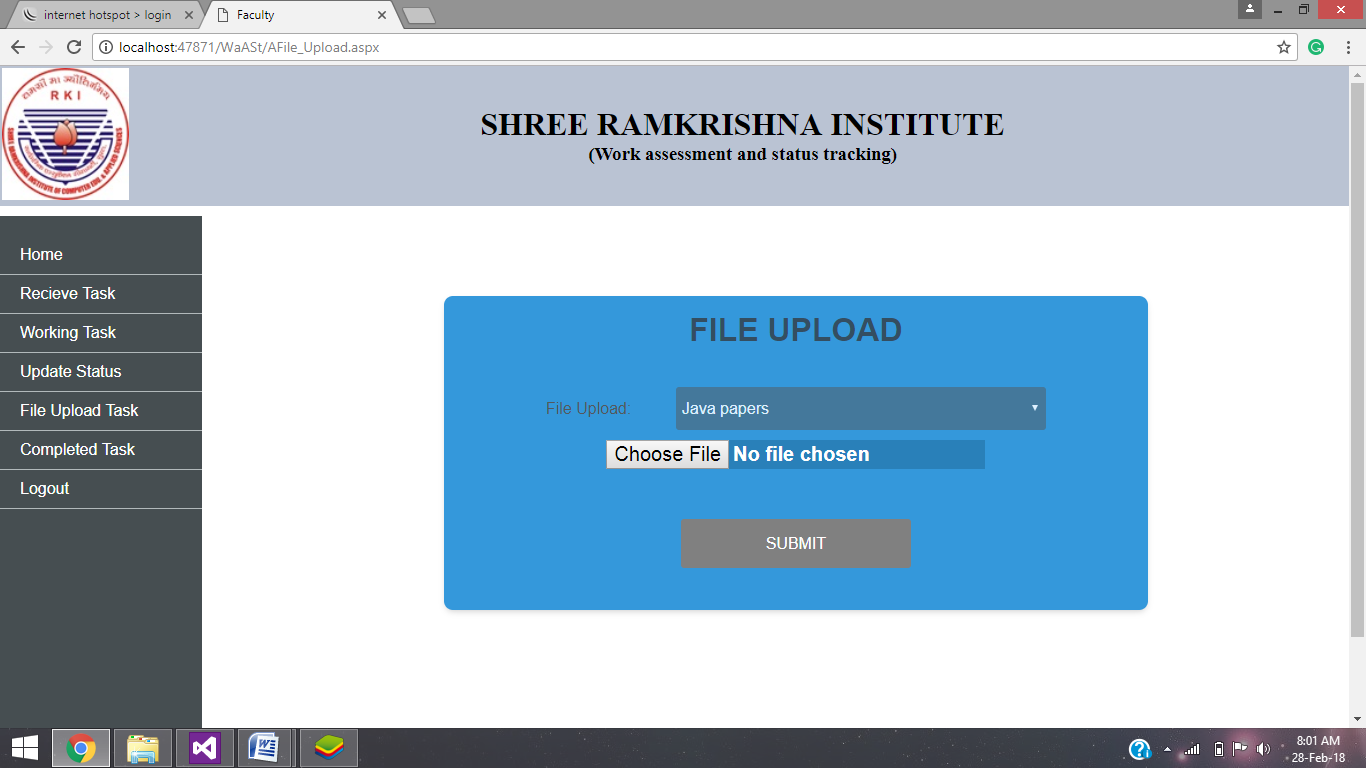
**Admin Page:-**

****

**Faculty Page:**

****

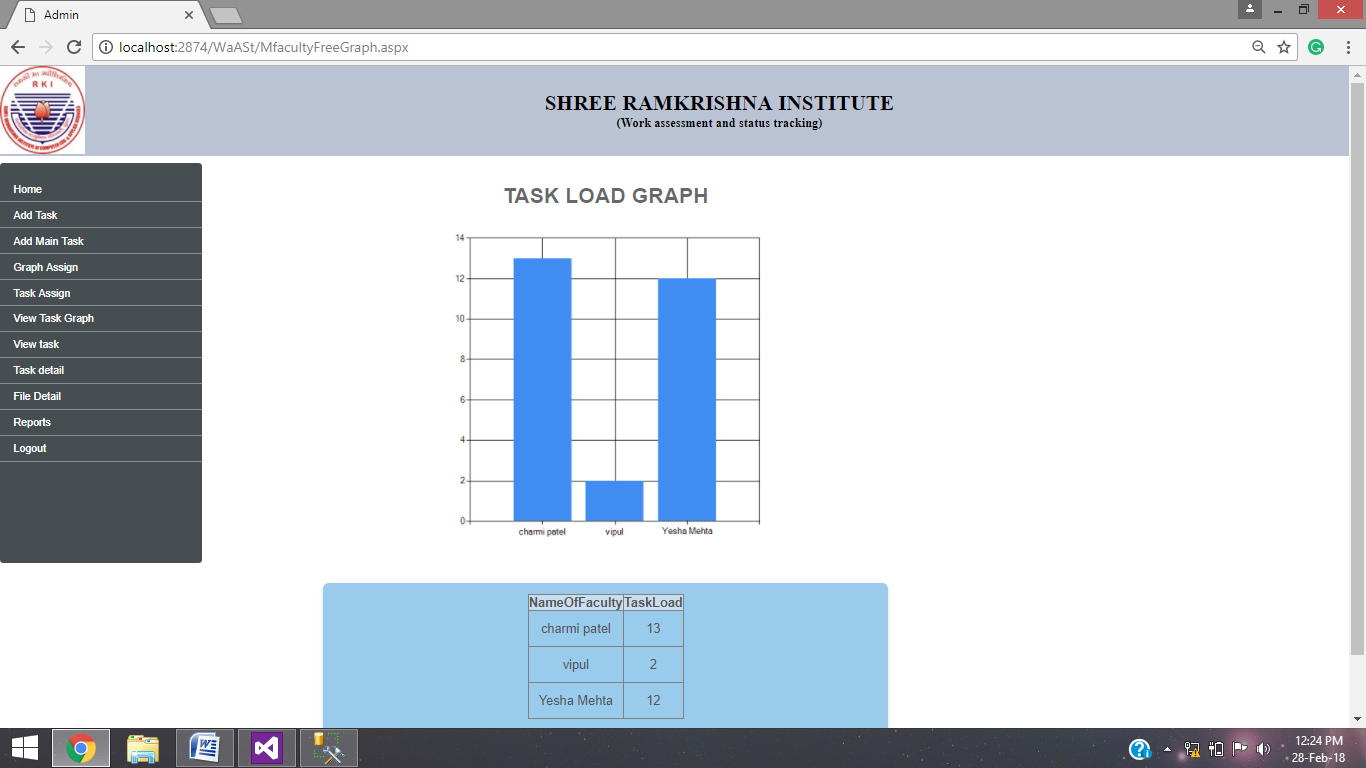
**File Upload Page:**

****

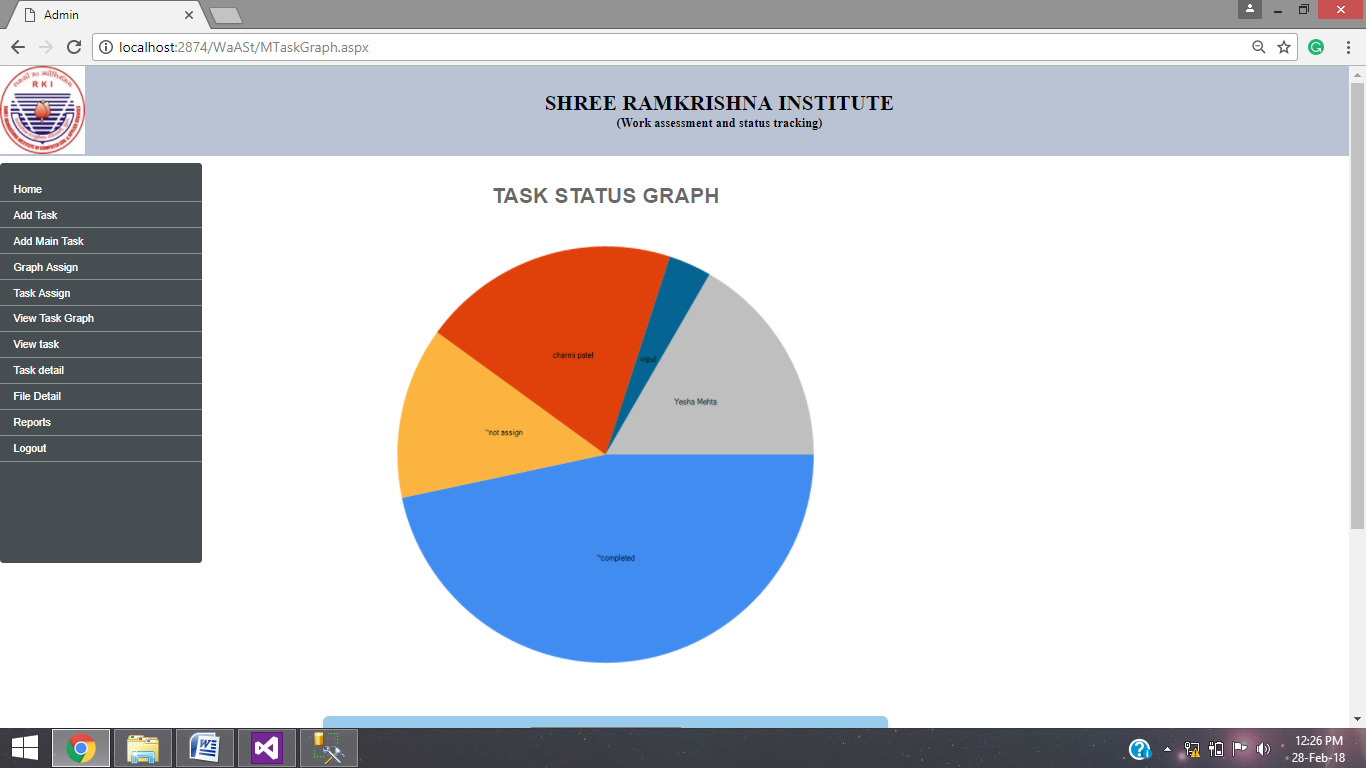
**Report Page:**

****

**Graph Assign Page:**

****

**Task Pie Chart Page:**

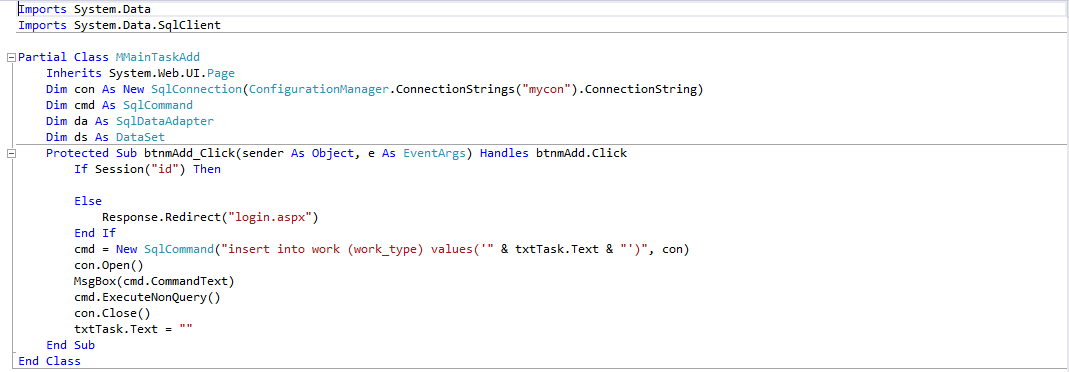
****

**5. CODING:**

**Login Page:**

****

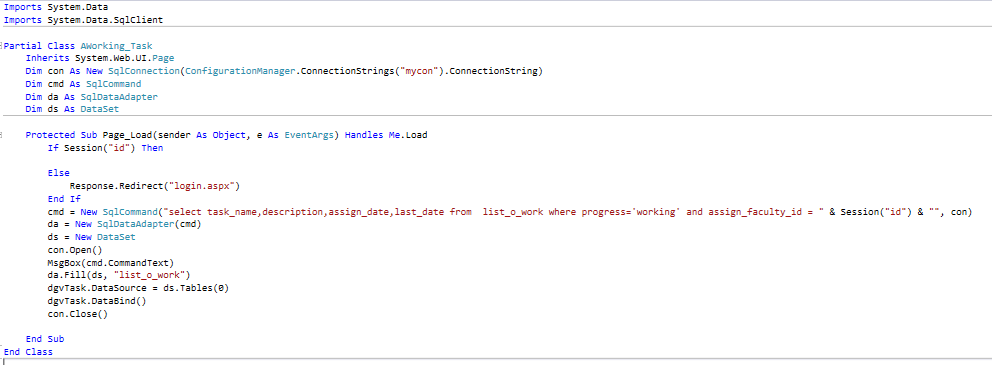
**Task Add Page:**



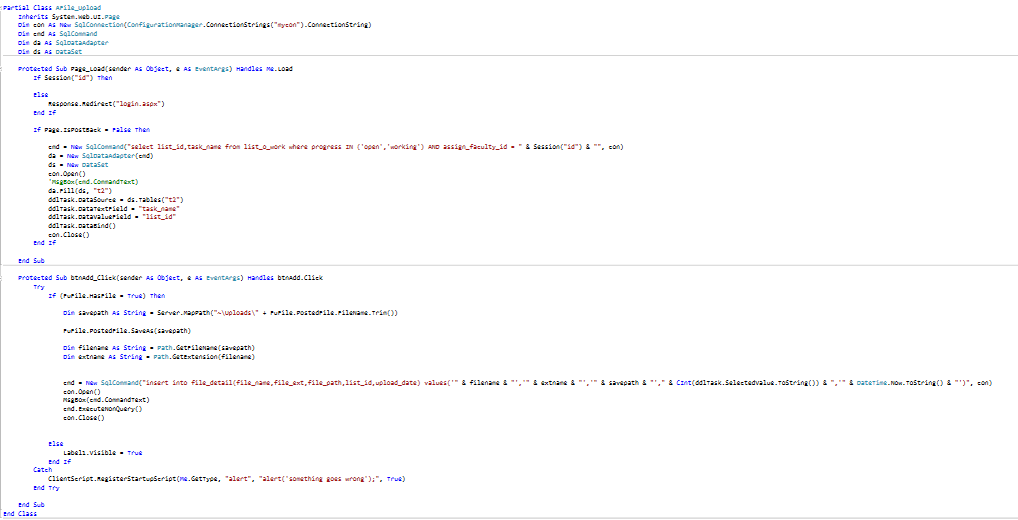
**E-mail Page:**

****

**View Page:**

****

**File Upload Page:**



**Report Page:**



**6. Testing**

* LOGIN PAGE:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| NO. | Input | Description | Expected Output | Actual Output | Result |
| 1 | Empty Username | No input | Error | Error | PASS |
| 2 | Empty Password | No Input | Error | Error | PASS |
| 3 | Wrong Username | Wrong username inputted | Login Failed | Login Failed | PASS |
| 4 | Wrong Password | Wrong password inputted | Login Failed | Login Failed | PASS |
| 5 | Valid data | Valid data entered | Login successful | Login successful | PASS |

* REGISTRATION PAGE:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| NO. | Input | Description | Expected Output | Actual Output | Result |
| 1 | Empty Name | Name not inputted | Error | Error | PASS |
| 2 | Empty Email | Email not inputted | Error | Error | PASS |
| 3 | Empty Contact | Contact not inputted | Error | Error | PASS |
| 4 | Empty Username | No Username | Error | Error | PASS |
| 5 | Empty Password | No Password | Error | Error | PASS |
| 6 | Valid data | Valid data entered | Error | Error | PASS |

* TASK ASSIGNMENT PAGE:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| NO. | Input | Description | Expected Output | Actual Output | Result |
| 1 | Empty Task List | Task not selected | Error | Error | PASS |
| 2 | Empty Faculty List | Task not selected | Error | Error | PASS |
| 3 | Start Date is greater than End Date | Assigning Date is inputted greater. | Error | Error | PASS |
| 4 | End Date is greater than Start Date | Submission Date is inputted greater. | Date Assigned for task | Date Assigned for task | PASS |
| 5 | Valid Data | Valid Data Entered | Task Assigned | Task Assigned | PASS |

* TASK ADD PAGE:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| NO. | Input | Description | Expected Output | Actual Output | Result |
| 1 | Empty Task Type List | Task Type not selected | Error | Error | PASS |
| 2 | Empty Task Name | Task Name not inputted | Error | Error | PASS |
| 3 | Task Type List  Entered | All Required Data Entered | Task Type Added | Task Added | PASS |
| 4 | Task Name  Entered | All Required Data Entered | Task Name added | Task Name added | PASS |
| 5 | All Data Entered Correctly | All Data Entered | Task is added | Task is added | PASS |

* FILE UPLOAD PAGE:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| NO. | Input | Description | Expected Output | Actual Output | Result |
| 1 | Empty Task Type List | Task Type not selected | Error | Error | PASS |
| 2 | No file selection | No file browsed or selected | Error | Error | PASS |
| 3 | Task Type selected | Task type selected | Task Type Added | Task Type Added | PASS |
| 4 | File Selected (Browsed) | File Browsed | File Name and File are fetched | File Name and File are fetched | PASS |
| 5 | All Data Entered Correctly | All Data selected and inserted | File Uploaded | File Uploaded | PASS |

**7. New Tools/ Technologies learned/used**

* JavaScript and CSS are used for web designing and, for dynamic web scripts.
* SQL Server Management studio is used for data storing by connecting to our website.
* Email API is used for sending email to the faculties, which will be done by the admin.
* JavaScript is used to disable “Back Button”, to avoid login back after user has logged out.
* Chart And Graph component is used to show graphs and charts.
* “iTextSharp” plugin is used to develop report from the system database and download it in multiple formats.

**8. System Limitations/Restrictions and Dependencies/Constraints**

* We cannot assign a single task to multiple people.
* We cannot assign multiple tasks to a single person at a time.

**9. Future Enhancement & Opportunities**

* More than one user can get access to a single task.
* Chat system can be used for conversation and better understanding of task.

**10. Bibliography & References**

* BOOKS
* Professional Asp.net 3.5
* ASP.NET for dummies
* The Complete Reference to ASP.NET
* WEBSITES
* [www.codeproject.com](http://www.codeproject.com)
* [www.tutorialspoint.com](http://www.tutorialspoint.com)
* [www.w3school.org](http://www.w3school.org)
* [www.github.com](http://www.github.com)
* [www.aspsnippet.com](http://www.aspsnippet.com)