**Project Report**

**on**

**Student Project Management System**

**Submitted By:**

Ramesh Maharjan

Binod Sharma

Shanta Acharya

Sushil Shrestha

**Submitted To:**

Asst.Prof. Bal Krishna Bal

DCGE

Kathmandu University

**Chapter 1**

**Introduction**

In general, project management software is a term covering and dealing with mainly estimation and planning, scheduling, cost control and budget management, resource allocation, communication, quality management and documentation of small and large projects.

The project, Student Project Management System (SPMS), is a simple web-based project management platform meeting all the criteria of project management software. Success of SPMS depends on the mutual interactions and contributions of students and supervisors equally.

Visually effective and simple GUI comforts SPMS users. All the necessary features such as adding and editing projects, tasks, files and folders, grades, calendar including other small features are present in SPMS. Emailing to concerned users of minor updates like addition and deletion of users is another important feature in SPMS. Leaving messages and replying to messages are also present including online chatting system. Every detail of events and actions are stored and displayed concerning particular user and projects. Authentication is the mandatory part of SPMS. Completion of tasks assigned and submission of deliverables and files causes projects progression which is displayed in an effective way. Budget and deadline management is also present is SPMS.

The key features are briefly explained below:

* **Project management:** projects can be added, edited by all members with some extra features provided to supervisors to open, close, and delete projects. Grades being the important feature of SPMS would be viewable by all members associated with graded projects with editing features for supervisors associated. Milestones, Tasks, and files being the major factors to show the progression of projects can be managed equally by students and supervisors associated with particular project.
* **Improve team collaboration:** emails of every minor changes and updates, online chatting and messaging features helps to share information and get updated in an instant which saves valuable time and effort and improve the interaction between students and supervisors to great extent.
* **Security and simplicity:** Administrator is the responsible member in SPMS dealing with management of users, both students and supervisors, and their respective features so there is no chance of misuse of SPMS. GUI is made as much simple as possible and the categorization of different features under tabs and sections makes the system easy to use.
* **Other features:** Frequent data backup facility ensures that no information is lost and easy search of recent and old project information. Tracking of project progress can be done with the help of calendar showing all the deadlines and important dates. Recent activities and logs can be seen as soon as logged in. files and folders can be downloaded as well for offline use.

Moreover, SPMS acts as a knowledge hub where members can share their information, ideas and knowledge related to project topic.

**Chapter 2**

**Background Theory**

This chapter contains the general approaches of developing a project management system and the essential features needed in the project management software. As described in the introduction part, project management software mainly deals with estimating, planning, scheduling and tracking of project with other extra features.

Generally speaking, scheduling and providing information are the key features of project management software and are described as below.

### Scheduling

One of the most common purposes is to schedule a series of events or tasks and the complexity of the schedule can vary considerably depending on how the tool is used. Some common challenges include:

* Events which depend on one another in different ways or dependencies.
* Scheduling people to work on, and resources required by, the various tasks, commonly termed resource scheduling.
* Dealing with uncertainties in the estimates of the duration of each task.

### Providing information

Project planning software can be expected to provide information to various people or stakeholders, and can be used to measure and justify the level of effort required to complete the project(s). Typical requirements might include:

* Tasks lists for people, and allocation schedules for resources
* Overview information on how long tasks will take to complete
* Early warning of any risks to the project
* Information on workload, for planning holidays
* Evidence (files, milestones, logs and activities)
* Historical information on how projects have progressed, and in particular, how actual and planned performance are related
* Optimum utilization of available resource

The following approaches can be followed to develop project management system.

### Desktop

Project management software can be implemented as a program that runs on the desktop of each user. This typically gives the most responsive and graphically-intense style of interface. Desktop applications typically store their data in a file, although some have the ability to collaborate with other users or to store their data in a central database. Even a file-based project plan can be shared between users if it's on a networked drive and only one user accesses it at a time.

Desktop applications can be written to run in a heterogeneous environment of multiple operating systems, although it's unusual.

### Web-based

Project management software can be implemented as a Web application, accessed through an intranet, or an extranet using a web browser.

This has all the usual advantages and disadvantages of web applications:

* Can be accessed from any type of computer without installing software on user's computer
* Ease of access-control
* Naturally multi-user
* Only one software version and installation to maintain
* Centralized data repository
* Typically slower to respond than desktop applications
* Project information not available when the user (or server) is offline
* Some solutions allow the user to go offline with a copy of the data

### Personal

A personal project management application is one used at home, typically to manage lifestyle or home projects. There is considerable overlap with single user systems, although personal project management software typically involves simpler interfaces.

### Single user

A single-user system is programmed with the assumption that only one person will ever need to edit the project plan at once. This may be used in small companies or ones where only a few people are involved in top-down project planning. Desktop applications generally fall into this category.

### Collaborative

A collaborative system is designed to support multiple users modifying different sections of the plan at once; for example, updating the areas they personally are responsible for such that those estimates get integrated into the overall plan. Web-based tools, including extranets, generally fall into this category, but have the limitation that they can only be used when the user has live Internet access. To address this limitation, some software tools using client–server architecture provide a rich client that runs on users' desktop computer and replicate project and task information to other project team members through a central server when users connect periodically to the network. Some tools allow team members to check out their schedules (and others' as read only) to work on them while not on the network. When reconnecting to the database, all changes are synchronized with the other schedules.

### Integrated

An integrated system combines project management or project planning, with many other aspects of company life. For example, projects can have bug tracking issues assigned to each project, the list of project customers becomes a customer relationship management module, and each person on the project plan has their own task lists, calendars, and messaging functionality associated with their projects.

So, after going through all the details and researched facts we concluded to develop our Student Project Management System (SPMS) as a web-based system. The E-R diagram of our system is as below:

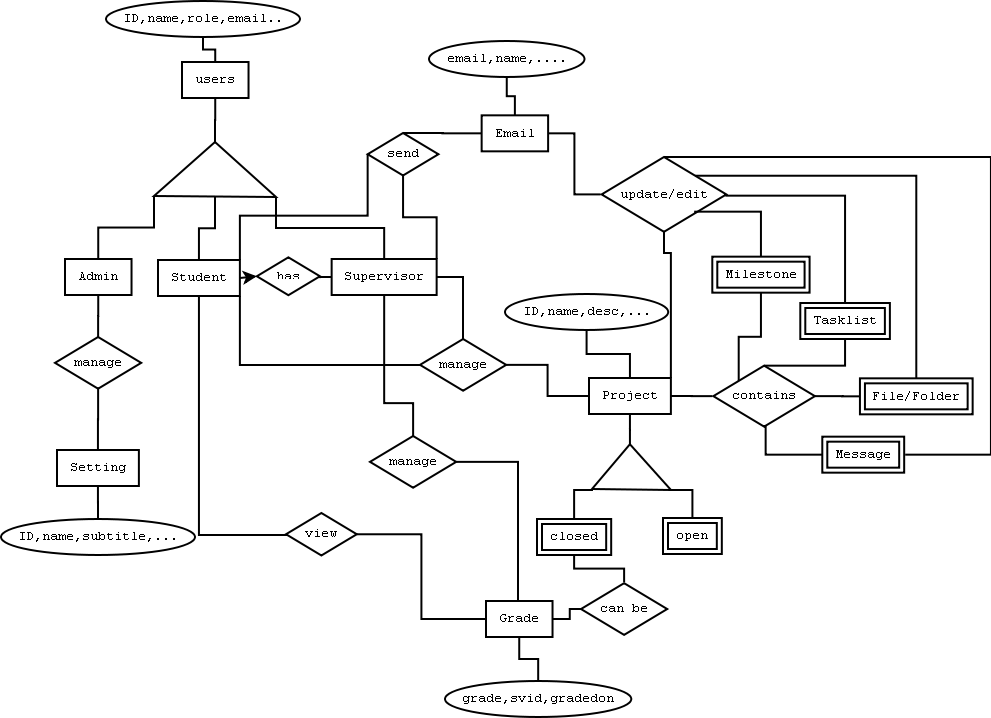


Fig 2.1 E-R diagram of SPMS

**Chapter 3**

**Challenges and problems**

Student Project Management System (SPMS) is in itself a challenging word and making the system interactive, attractive and user friendly is a tough job. This chapter describes the challenges and problems faced in the design and implementation of SPMS, which has been done by dividing the works into phases with different difficulties encountered: research phase, coding, debugging and testing phase and finally implementation phase with debugging.

**3.1 Research phase:**

Choosing a platform for designing the project was difficult as there were many project management systems developed in different platforms. After a long research, web-based project management system with PHP as the programming language was the choice as there were many resources available. And, knowledge of web-server and database was mandatory.

**3.2 Coding, debugging and testing phase:**

Since most of the codes could be easily available due to open-source facility, less time was spent on creating new codes. But the unfamiliar available codes needed a lot of time and many other resources were needed to be consulted in order to understand and apply them in the project. With time codes became so massive that it was difficult to follow and understand the links and flows inside the project. Most of the testing and debugging were done using local server, so networking bugs and errors were fully avoided but with minor errors with local server and database. Email system has to be implemented for every minor facility available in SPMS and it can be seen that there are many facilities available.

**3.3 Implementation phase:**

Implementation had been module as small parts in the coding, debugging and testing phase. This phase mainly includes the task of testing the project on different servers.

Challenges and problems faced by our group members while designing SPMS would be more evident in the next chapter where parts of the project is defined in brief.

**Chapter 4**

**Parts of SPMS**

This chapter briefly describes the parts of the project SPMS (Student Project Management System). Different people with different roles and authorities interact with each other for different projects with different milestones, tasks and task-lists, messages, files and folders and grades assigned to them in the SPMS system.

**4.1 Roles and users**

Initially there are only three roles available: administrator, supervisor and student. Administrator has authority to add, delete and edit users, roles, projects, milestones, tasks and all the other facilities available in SPMS. This account is also responsible for registering new users either as supervisor or student. Supervisor is equivalent in authority with administrator except that this account is not authorized to access settings. Student account is not authorized to open and close a project, post, edit and delete grades with some other minor restrictions.

Adding new roles feature is also available in SPMS.

**4.2 Desktop**

The home page of every account when logged-in is called desktop in SPMS. This page displays all the added open projects, tasks, messages and activities assigned and related with logged account with calendar, search and chat options. Projects, tasks and messages can be added and edited if displayed in desktop.

**4.3 My tasks**

This section displays all the pending task-lists and tasks associated with the current account.

**4.4 My messages**

This section displays all the messages related with the current account.

**4.5 My grades**

This section displays the grades of graded projects. Students can only view it but supervisor has authority to edit it as well.

**4.6 My projects**

All the open and closed projects can be viewed in this section. Projects can be added, opened and closed in this section. Clicking on the project leads to individual project leads to more options inside individual projects where a member can edit projects, edit milestones, task-lists, tasks, messages, upload and edit files and folders and if supervisor then edit grades.

i think this part can be included in the conclusion part:

To sum up, SPMS consists of all the necessary features of a typical student project management system with facilities to manage projects, files and folders, milestones, tasks, messages, users and grades. Moreover the system even notifies the minor changes occurred in the account and projects associated to related users through email.

**Chapter 5**

**Database and system**

Database is the most important part of SPMS as it stores all the information about users, projects, activities, files and folders, grades and other information. This chapter defines the core parts of the database system and the interaction of SPMS with database. Necessary screenshots has been taken for visual illustrations. There are altogether 17 tables in the database: settings, user, tasklist, tasks, tasks\_assigned, roles, roles\_assigned, projekte, projekte\_assigned, files, files\_attached, projectfolders, milestones, milestones\_assigned, messages, log, and chat as can be seen in the following screenshot.

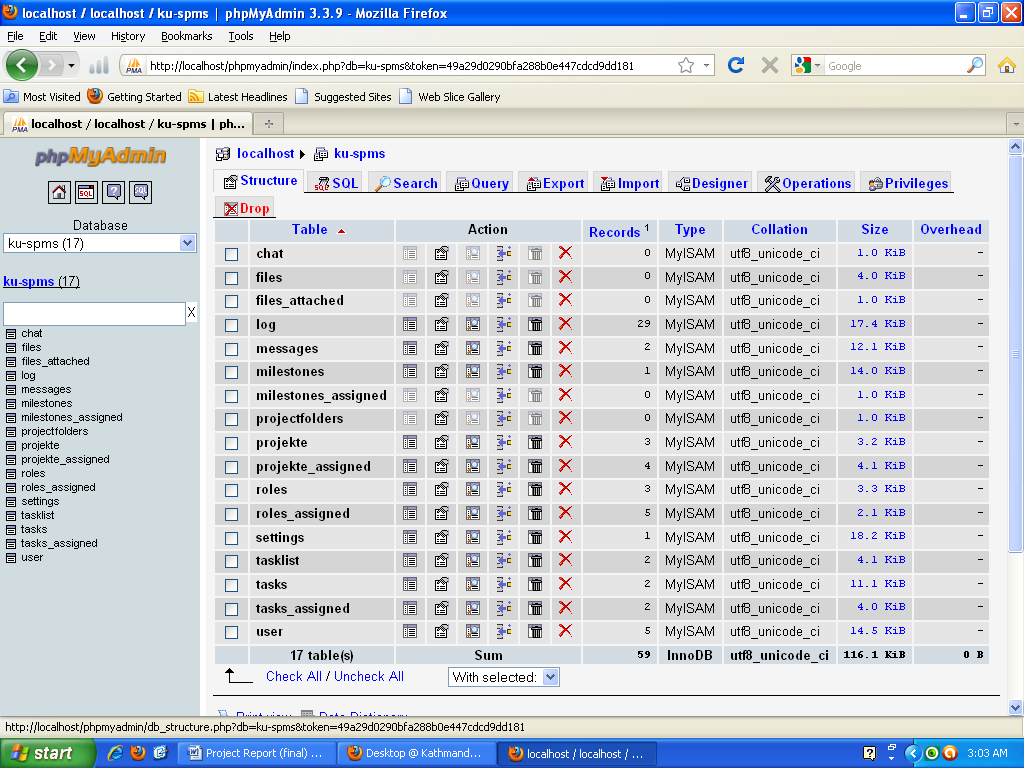


Fig 5.1 database tables of KU-SPMS

5.1 User

User table consists information about users added (either supervisor or student or Admin) in KU-SPMS. The primary key is ID and other information would be clear with the following screenshot. Email id is used to send emails to the users.

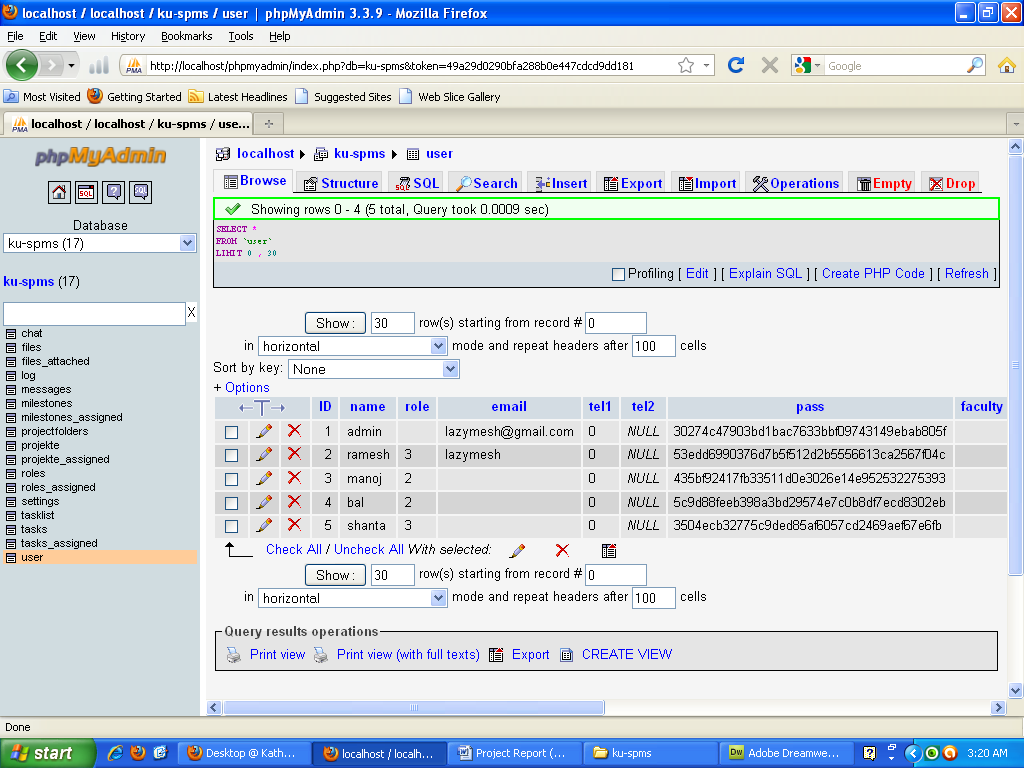


Fig 5.2 User table in KU-SPMS

5.2 Settings

Settings table consists of information about the system such as name of the system, mailing server name and password and many more. The screen shot is as below:

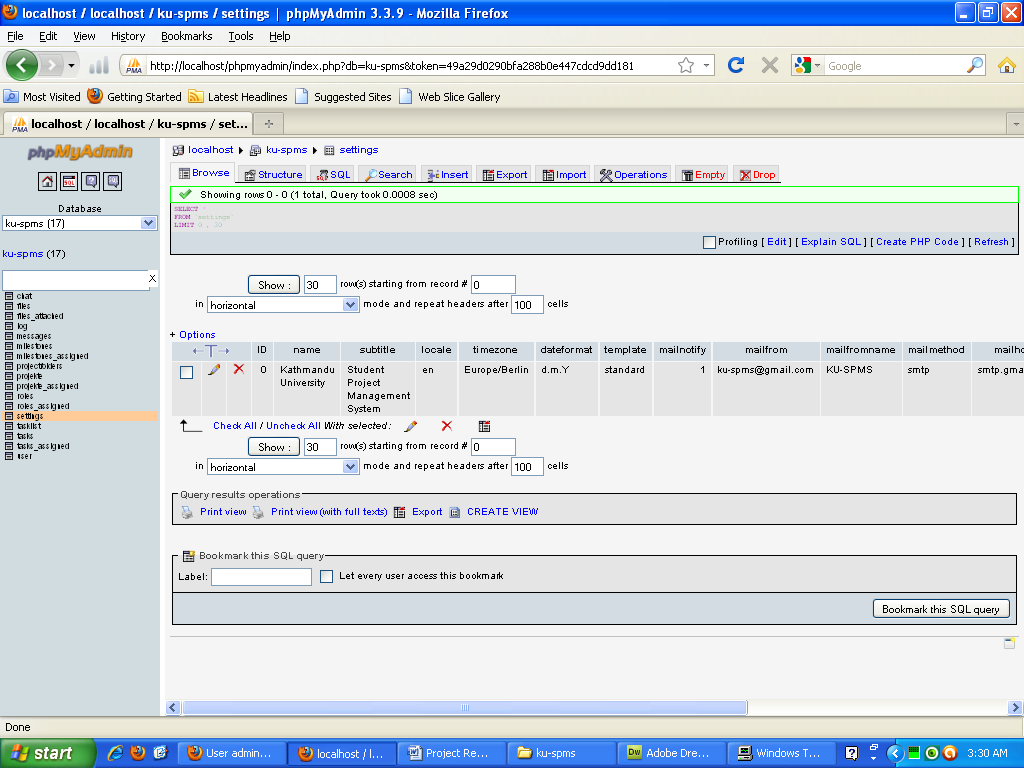


Fig 5.3 Settings table in KU-SPMS

5.3 Projekte and projekte\_assigned

Projekte table contains all the necessary information regarding projects. Grade has been embedded inside this table for better results. The primary key is ID. Projekte\_assigned table contains information about the assignment of projects to users. The primary key for this table is also ID. The screenshots are given below.

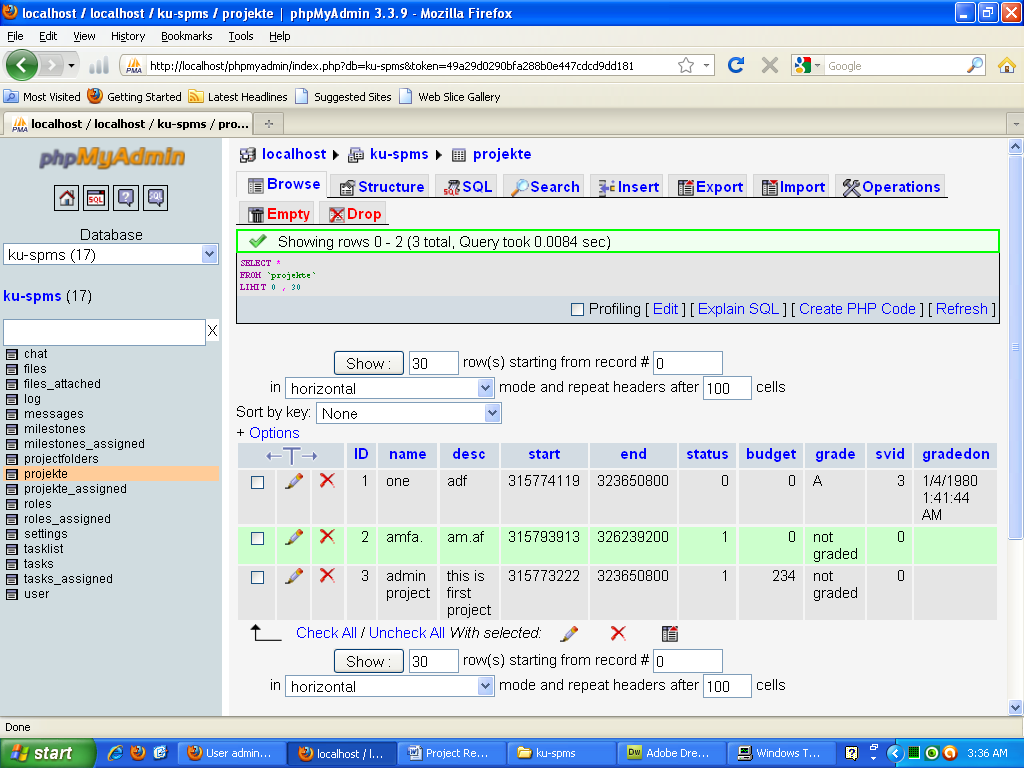


Fig 5.4 Projekte table in KU-SPMS

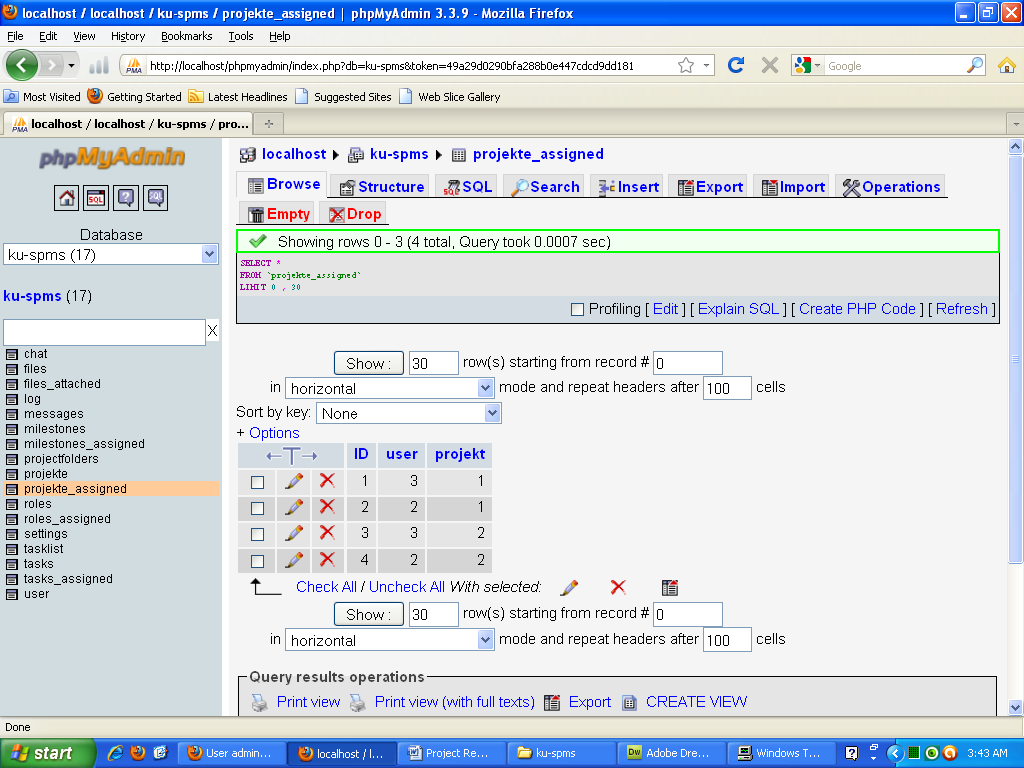


Fig 5.5 Projekte\_assigned table in KU-SPMS

5.4 Roles and roles\_assigned

Roles table contains data about the roles created and the permission allowed for each roles created. KU-SPMS’s default roles are Admin for creating and managing users and settings, students and supervisors for managing projects and other related tasks. Admin can create another role if needed. The primary key is ID.

Roles\_assigned table consists information about the assignment of roles to users i.e. the users roles as student or as supervisor are stored in roles\_assigned table. The primary key is ID.

Screenshots for both tables are given below:

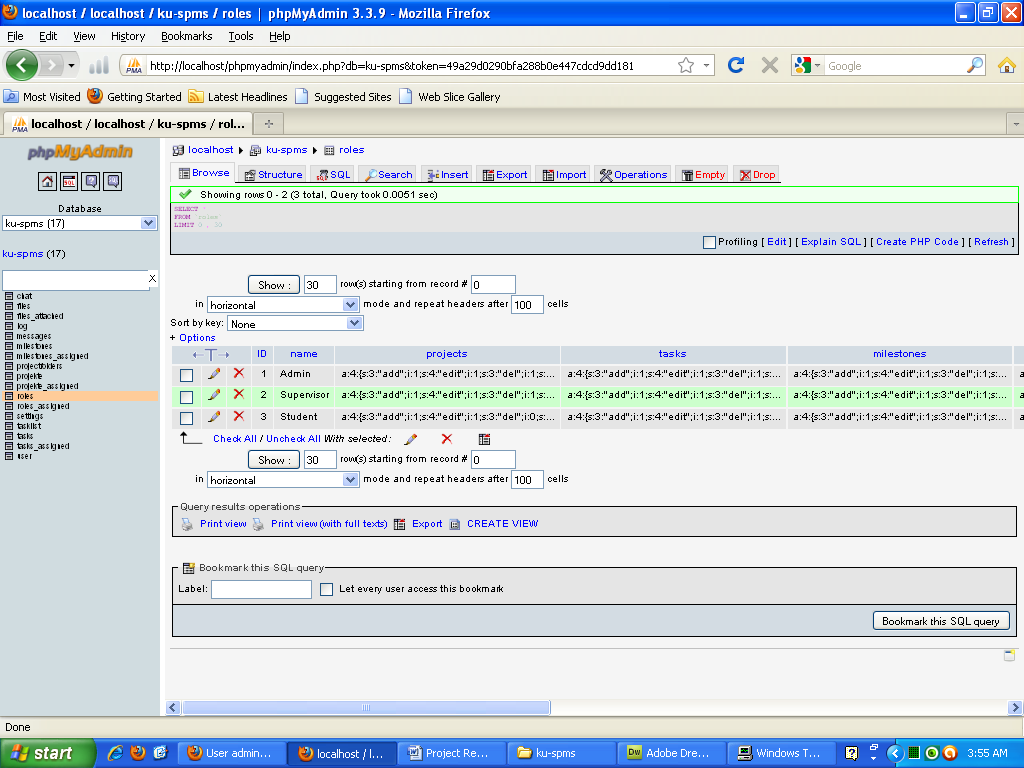


Fig 5.6 Roles table in KU-SPMS

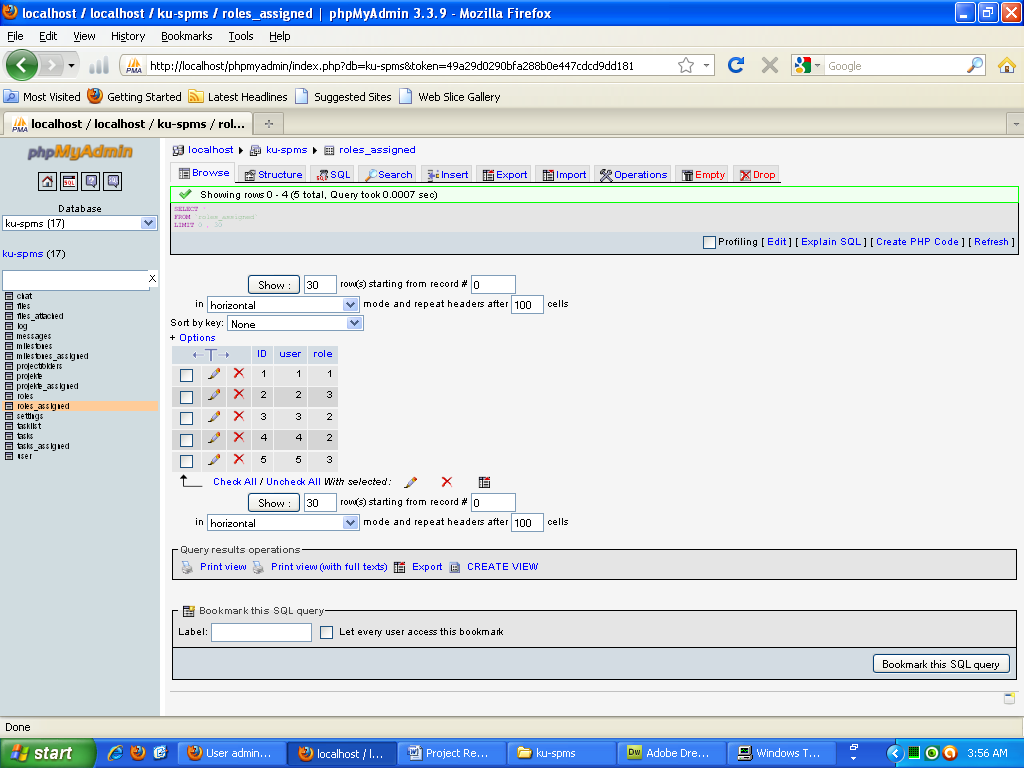


Fig 5.7 Roles\_assigned table in KU-SPMS

5.5 Milestones, Tasklist, Tasks, Tasks\_assigned

Milestones table contains information about milestones created with its deadline and associated project with other information. Tasklist table contains information about task-list if created any. A task can be created only after a task-list has been opened. Tasks table stores information about the assigned tasks with its submission dates. Tasks\_assigned table is used to store the ids of users associated with particular tasks.

All of these tables store information about the deliverables and tasks that a student has to submit on time in order to achieve good grades. The screen shots would clear the ideas further.

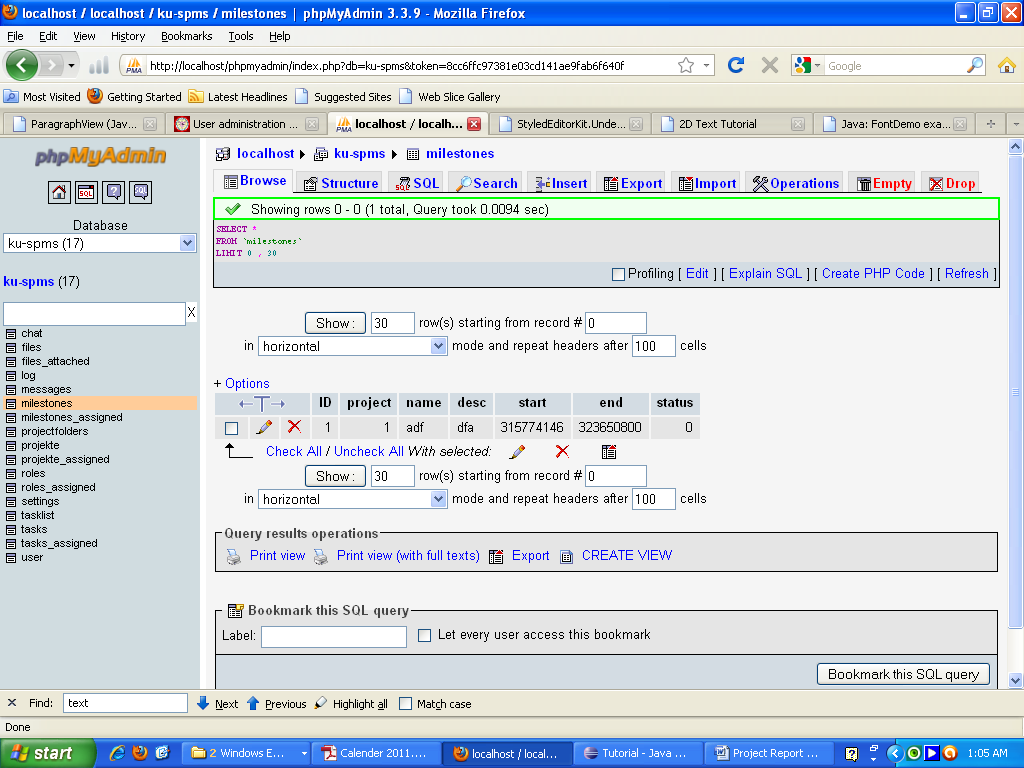


Fig 5.8 Milestones table in KU-SPMS

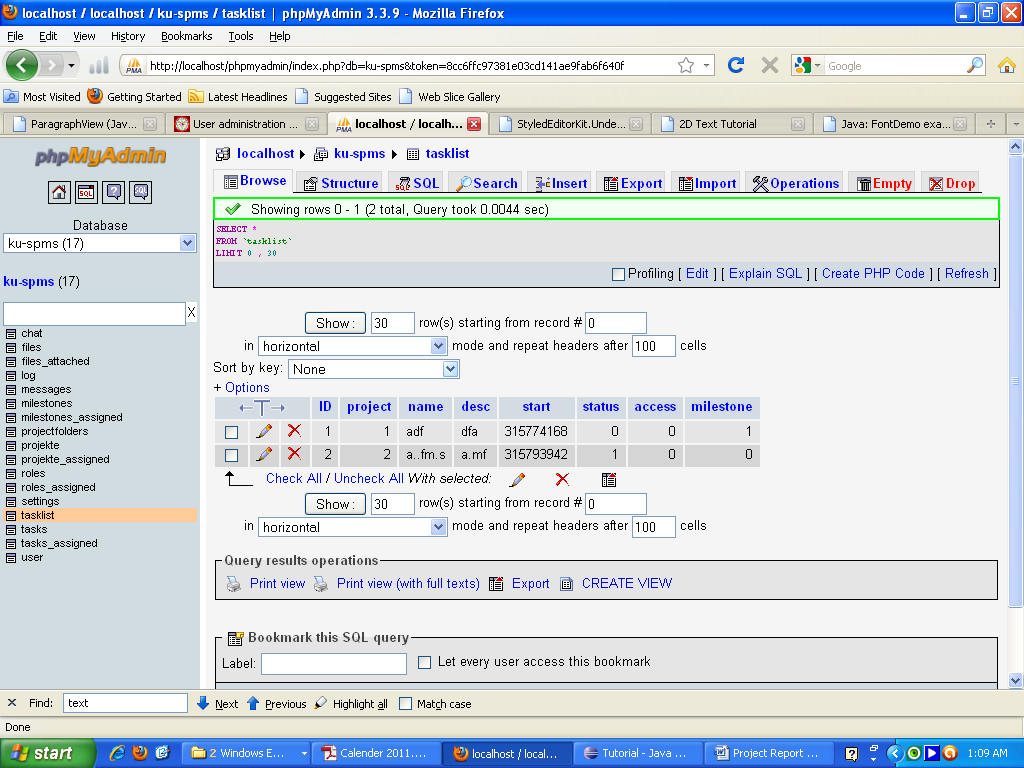


Fig 5.9 Tasklist table in KU-SPMS

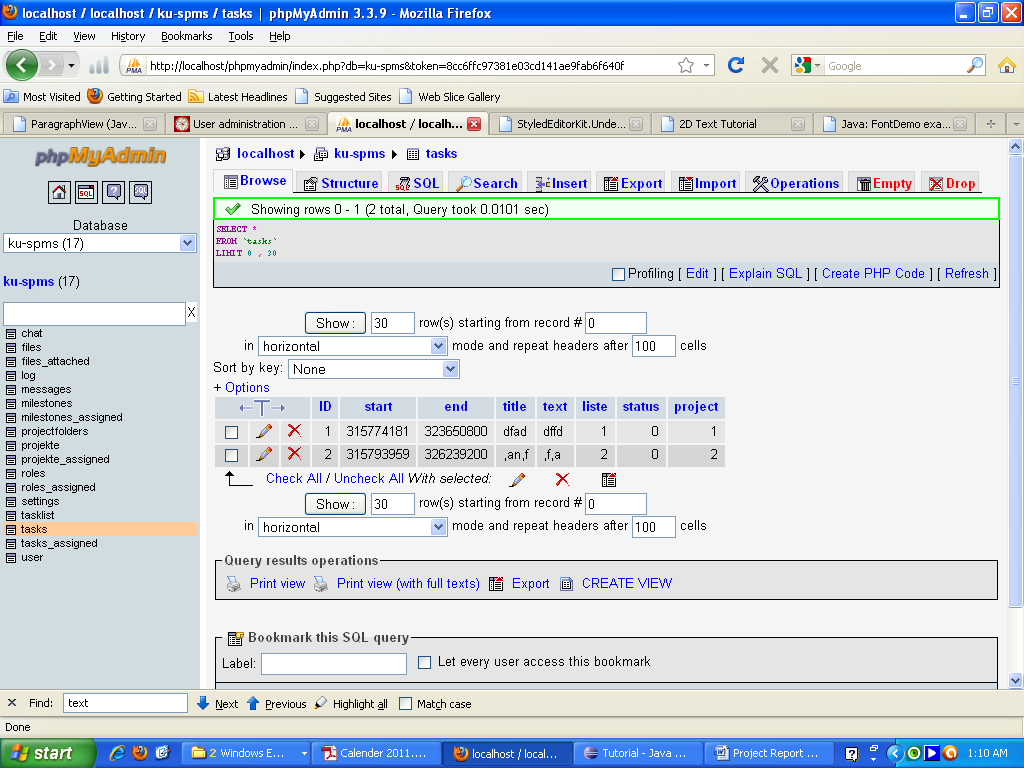


Fig 5.10 Tasks table in KU-SPMS

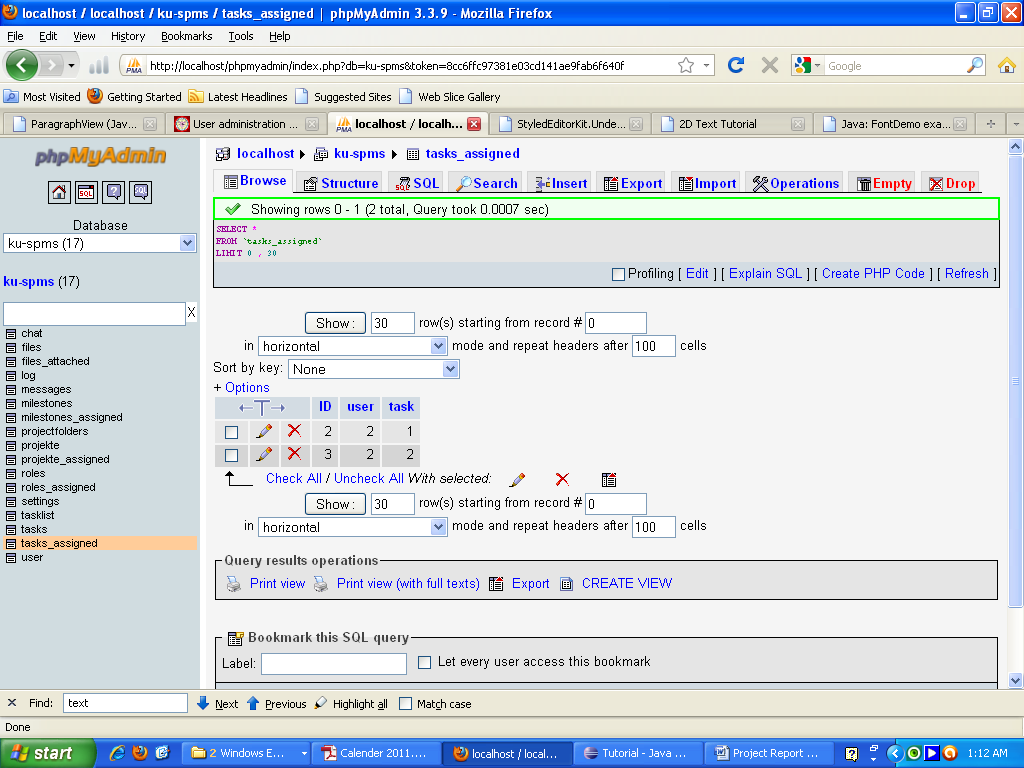


Fig 5.11 Tasks\_assigned table in KU-SPMS