FINAL Project report

Course :cse311(Database management system)

PROJECT NAME: ICT division innovation grant commission for fellowship management system

Section: 12

Semester :summer 2024

Faculty Initials: Slf

Submission date: 02-12-2024

Group number 03

Member details:

Md.Touhiduzzaman Saiem 2233356642

Rokibul Islam 2111088642

As part of the digital transformation initiative for the ICT Division innovation grant commission for fellowship management system, we developed a **Fellowship Management System** designed to streamline the entire fellowship lifecycle, from application submission to project completion. This project was driven by the need for a more transparent, efficient, and user-friendly process for both applicants and administrators. Through this report, we will provide a comprehensive overview of each section of the system, including its functionalities, structure, and technical implementation.

Project Concept and Objective

The primary objective of the Fellowship Management System is to automate and enhance the fellowship process. Before this system, managing fellowships involved manual processes, paper-based applications, and frequent delays in communication. Our mission was to eliminate these inefficiencies by creating a centralized web-based platform that integrates all aspects of fellowship management. The system caters to three key user groups: applicants, mentors, and administrators, each with a tailored interface and functionality.

User Interface and Design

The user interface is designed to provide a clean, intuitive experience that guides users through the system's various processes. We prioritized simplicity and accessibility, ensuring that users, regardless of their technical expertise, can navigate the system with ease. The homepage features a navigation bar with links to essential sections, including the application form, mentorship resources, project tracking, and administrative tools. Each page is structured with a clear hierarchy of information, using HTML for structure and CSS for styling.

In the application form section, we implemented clear input fields with labels and placeholder text to guide users. For instance, the "Full Name" field prompts users with an example format, while the "Date of Birth" field includes a calendar picker for easy selection. Error messages appear in real-time if any required field is left empty or contains invalid data, enhancing user experience and reducing submission errors.

Application Submission Process

The core of the Fellowship Management System is its **application submission process**. Applicants start by registering with their email and creating a secure password. Once logged in, they access a multi-step application form that captures detailed personal and professional information. The form is divided into logical sections:

In the **Personal Information** section, applicants provide essential details such as full name, email address, phone number, and date of birth. These fields are validated to ensure data accuracy, with mandatory fields marked and error prompts guiding users to correct mistakes.

The **Educational Background** section allows applicants to input their academic qualifications, including degree, institution, and year of graduation. We used dropdown menus for fields like graduation year to maintain consistency and prevent errors.

In the **Professional Experience** section, applicants can list relevant work experience, internships, or projects. We provided a rich-text input field for detailed descriptions, with the option to add multiple entries, making the system flexible and comprehensive.

After completing the form, applicants review their information on a summary page before final submission. This ensures they can verify and edit their details, minimizing errors and incomplete submissions.

Application Status Tracking

Once an application is submitted, applicants can track their progress through the system's **status tracking feature**. Each application moves through several stages:

- **Submitted**: Confirms that the application has been received by the system and is awaiting review.
- **Under Review**: Indicates that administrators are currently assessing the application.
- **Approved**: Notifies the applicant of their successful selection as a fellow.
- **Rejected**: Informs the applicant if their application did not meet the criteria.

This real-time tracking system keeps applicants informed without requiring manual follow-ups, fostering transparency and reducing administrative workload.

Mentorship and Resource Management

Upon approval, fellows gain access to the **mentorship management module**, which is critical for their professional development. This section includes several key features:

Administrators can handle **Mentor Assignment** by pairing fellows with mentors based on their field of study and project requirements. Each mentor's profile includes their expertise and availability, ensuring appropriate matching.

Through **Mentor-Fellow Interaction**, fellows can view their mentor's contact information and schedule meetings via the system. A built-in messaging tool facilitates secure and efficient communication between fellows and mentors.

The **Resource Sharing** feature allows mentors to upload guides, reading materials, and other resources. Fellows access these materials through their dashboard, creating a centralized repository for all learning resources.

Project Management Module

A distinguishing feature of this system is the **project management module**, which supports fellows throughout their project lifecycle. This module is tightly integrated into the mentorship process to enable collaborative progress tracking between fellows and mentors.

Milestone Tracking allows fellows to set and manage project milestones, each associated with a due date and specific deliverables. This structured approach ensures steady progress.

Through **Deliverable Submissions**, fellows upload project files categorized by milestone, making it easier for mentors to review and provide feedback.

The **Feedback Mechanism** lets mentors leave comments and suggestions on submitted deliverables. Fellows receive notifications when feedback is provided, enabling continuous improvement and timely revisions.

Technical Implementation

The Fellowship Management System is built using a combination of **HTML, CSS, PHP, and MySQL**, ensuring a robust and scalable platform.

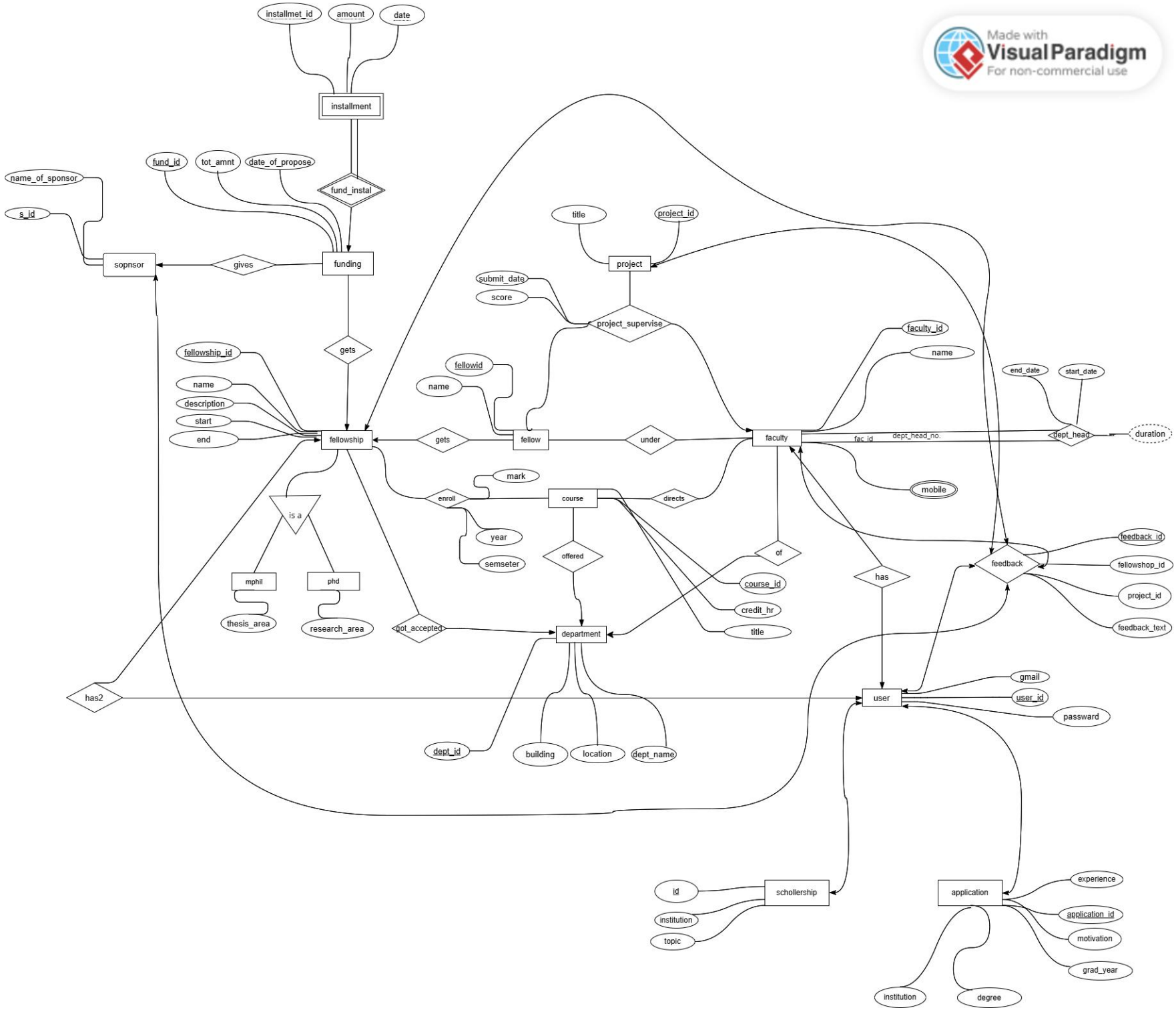
The frontend employs HTML for structuring content and CSS for visual presentation, creating a responsive and user-friendly interface.

The backend, developed in PHP, handles data processing and business logic. For instance, user input is processed through PHP scripts that validate and sanitize the data before storing it in the

MySQL database. The database is designed with a normalized schema, ensuring data integrity and efficient querying. Key tables include:

- The **Users** table stores login credentials and profile information for applicants, mentors, and administrators.
- The **Applications** table contains details of submitted applications, their current status, and timestamps.
- The **Mentorships** table tracks mentor-fellow pairings and associated project information.

Project Erd and schema diagram are given in following pages :


installmet_id
amount
date
installment
fund_id
tot_amnt
date_of_propose
fund_instal
name_of_sponsor
s_id
sopnsor
gives
funding
gets
fellowship_id
name
description
start
end
fellowship
title
project_id
project
submit_date
score
project_supervise
fellowid
name
gets
fellow
under
faculty
faculty_id
name
end_date
start_date
dept_head
duration
fac_id
dept_head_no.
mobile
mark
enroll
course
directs
year
semseter
offered
of
course_id
credit_hr
title
is a
mphil
phd
thesis_area
research_area
not_accepted
department
has
feedback
feedback_id
fellowshop_id
project_id
feedback_text
has2
user
gmail
user_id
passward
dept_id
building
location
dept_name
schollership
id
institution
topic
application
experience
application_id
motivation
grad_year
institution
degree

Relation schema of the project

1) Fellow(**fellow_id**,name,fellowship_id,user_id)
2) Faculty(**fac_id,** name, dept_id,user_id)
3) Fellowship(**fellowship_id,** name, description,start,end,dept_id, user_id)
4) Course(**course_id,** title, credit_hr,dept_id)
5) User(**user_id,** name, email,pass)
6) Sponsor(**s_id,**name_of_sponsor,user_id)
7) Mphilfellowship(**fellowship_id**, thesis_area)
8) Phdfellowship(**fellowship_id**, research_area)
9) Department(**department_id,**dept_name, location, building)
10) Funding(**fund_id**, tot_amnt,date_of_submit,fellowship_id)
11) Installment(**fund_id,installment_id,amount,date**)
12) Project(**project_id,** project_title)
13) Project_Supervise**(project_id,faculty_id, fellow_id**,marks)
14) Enroll(**course_id, fellowship_id,year,semester**,mark)
15) Facultymobile(**faculty_id, mobile**)
16) Dept_head(**dept_head_no.,faculty_id,start,end**)
17) Applications(**application_id**,u_id,degree, institution,grad_year,experience , motivation)
18) Schollership_application(**id,name, institute** ,topic,major)
19) Feedback(**feedback_id**,fellowship_id,project_id,feedback_text,created_at)

CREATE TABLE applications:

```
(
    application_id INT PRIMARY KEY AUTO_INCREMENT,
    u_id INT NOT NULL,
    degree VARCHAR(255) NOT NULL,
    institution VARCHAR(255) NOT NULL,
    graduation_year INT NOT NULL,
    experience TEXT,
    motivation TEXT
);
```

CREATE TABLE courses:

```
(
    course_id INT PRIMARY KEY AUTO_INCREMENT,
    title VARCHAR(255) NOT NULL,
    credit_hour INT NOT NULL
);
```

CREATE TABLE department:

```
(
    dept_id INT PRIMARY KEY AUTO_INCREMENT,
    dept_name VARCHAR(255) NOT NULL,
    building VARCHAR(255),
    location VARCHAR(255)
);
```

CREATE TABLE faculty:

```
(
```

```
    faculty_id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(255) NOT NULL,
    mobile VARCHAR(15),
    start_date DATE,
    end_date DATE,
    dept_head BOOLEAN,
    dept_id INT NOT NULL,
    password VARCHAR(255) NOT NULL
);
```

CREATE TABLE fellow:

```
(
    fellow_id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(255) NOT NULL,
    fellowship_id INT NOT NULL
);
```

CREATE TABLE fellowship:

```
(
    fellowship_id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(255) NOT NULL,
    description TEXT,
    start DATE,
    end DATE,
    fund_id INT,
    course_id INT,
    u_id INT
);
```

CREATE TABLE funding:

```
(
    fund_id INT PRIMARY KEY AUTO_INCREMENT,
    tot_amt FLOAT NOT NULL,
    date_of_proposal DATE,
    phd_id INT
);
```

CREATE TABLE installment:

```
(
    id INT PRIMARY KEY AUTO_INCREMENT,
    fellow_id INT NOT NULL,
    amount FLOAT NOT NULL,
    due_date DATE,
    created_at DATETIME DEFAULT CURRENT_TIMESTAMP
);
```

CREATE TABLE mph:

```
(
    id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(255) NOT NULL,
    email VARCHAR(255) NOT NULL,
    research_topic VARCHAR(255),
    created_at DATETIME DEFAULT CURRENT_TIMESTAMP
);
```

CREATE TABLE phd:

```
(
```

id INT PRIMARY KEY AUTO_INCREMENT,

name VARCHAR(255) NOT NULL,

email VARCHAR(255) NOT NULL,

field_of_study VARCHAR(255),

institution VARCHAR(255),

created_at DATETIME DEFAULT CURRENT_TIMESTAMP

);

CREATE TABLE project:

(

project_id INT PRIMARY KEY AUTO_INCREMENT,

title VARCHAR(255) NOT NULL,

start_date DATE,

end_date DATE,

duration INT,

fellowship_id INT NOT NULL

);

CREATE TABLE scholarship_applications:

(

id INT PRIMARY KEY AUTO_INCREMENT,

name VARCHAR(255) NOT NULL,

institute VARCHAR(255) NOT NULL,

topic VARCHAR(255),

major_in VARCHAR(255),

email VARCHAR(255),

phone VARCHAR(15),

u_id INT

);

CREATE TABLE sponsor:

(

s_id INT PRIMARY KEY AUTO_INCREMENT,

name_of_sponsor VARCHAR(255) NOT NULL

);

CREATE TABLE users:

(

u_id INT PRIMARY KEY AUTO_INCREMENT,

name VARCHAR(255) NOT NULL,

email VARCHAR(255) NOT NULL,

phone VARCHAR(15),

district VARCHAR(255),

department VARCHAR(255),

password VARCHAR(255) NOT NULL

);

CREATE TABLE feedback:

(

feedback_id INT PRIMARY KEY AUTO_INCREMENT,

fellowship_id INT,

project_id INT,

feedback_text TEXT,

created_at DATETIME DEFAULT CURRENT_TIMESTAMP,

feedback_source VARCHAR(255),

source_id INT

);

1. Admin Dashboard for User and Application Management

- **Feature**: Create an admin dashboard where admins can:
  - View all user registrations and applications.
  - Approve or reject fellowship applications.
  - View, edit, or delete user information.
- **Backend**: Add endpoints for fetching, updating, and deleting user and application data.
- **Database**: Include a table for `admin` roles with secure authentication, or add a `role` column in the `users` table to differentiate between users and admins.

2. Email Notifications for Application Status

- **Feature**: Send automated email notifications to users when:
  - They sign up.
  - Their application status changes (e.g., "Accepted" or "Rejected").
- **Backend**: Use PHP's `mail()` function or a library like PHPMailer to send emails.
- **Database**: Add an `application_status` log table to track changes, or store notification status in the `applications` table.

3. Enhanced User Profile and Settings

- **Feature**: Allow users to update their profile information and reset their password.
- **Backend**: Create endpoints for updating profile data and resetting passwords.
- **Database**: Extend the `users` table to include optional fields, like `bio`, `profile_picture`, and `preferences`.

4. Application Tracking and Feedback

- **Feature**: Allow users to track their application progress in real-time and leave feedback on the fellowship program if accepted.
- **Backend**: Develop a system for updating application progress with custom stages (e.g., "Submitted," "Under Review," "Interview," "Final Decision").
- **Database**: Add a `status_log` table to record each application's stage with timestamps, or a `feedback` table for users' comments after program completion.

5. Mentor Matching System

- **Feature**: Implement a matching system that pairs accepted fellows with mentors based on expertise and interests.
- **Backend**: Develop logic to match users to mentors by shared skills or goals, with an option for mentees to select or request a mentor.
- **Database**: Expand the `mentors` table to include expertise areas, and add a `mentor_assignments` table that records fellow-mentor pairs.

6. Admin Analytics and Reports

- **Feature**: Provide an admin interface that displays analytics on user registrations, application status, and user engagement.
- **Backend**: Create endpoints for fetching aggregate data (e.g., total applications, acceptance rates).
- **Database**: Create aggregate views in SQL for faster data retrieval and summaries, or a `logs` table to store events and interactions.

7. Two-Factor Authentication (2FA) for Enhanced Security

- **Feature**: Implement 2FA using email or SMS during user login for added security.
- **Backend**: Integrate a 2FA service or create a 2FA code generator with email/SMS sending.
- **Database**: Store a temporary `2fa_code` and `2fa_expiry` in the `users` table to validate user identity upon login.

8. Audit Logs for User and Admin Actions

- **Feature**: Track actions performed by users and admins for accountability, especially useful in sensitive systems.
- **Backend**: Create backend logging functions that capture actions such as logins, profile updates, and application changes.
- **Database**: Add an `audit_logs` table to capture action details, timestamps, and user IDs associated with each action.

Implementing these features will create a robust, feature-rich backend and database that can support advanced functionalities and security. Let me know which ones interest you, and I can help with implementation details!

conclussion

Throughout development, we encountered several challenges. Ensuring accurate **data validation** was one of the primary issues, as incorrect data entries were initially common. We resolved this by implementing both client-side and server-side validation, ensuring a seamless and error-free submission process.

Designing an intuitive user interface for different user roles also presented a challenge. We conducted extensive user testing to identify pain points, which led us to refine the navigation structure and add contextual tooltips and guidance prompts.

While the system meets current requirements, there is potential for further enhancements. We plan to implement **automated email notifications** for status updates and upcoming deadlines. Additionally, a **mobile application** would enhance accessibility, particularly for users managing their applications on-the-go.

Developing the Fellowship Management System was a comprehensive and fulfilling project. By integrating various functionalities into a unified platform, we created a system that not only streamlines administrative tasks but also enhances the user experience for applicants and mentors. This project lays the foundation for future digital initiatives within the ICT Division, and we are excited to witness its long-term impact.