CSE334

Database Project

TuitionPlatform

August 26,2023

Overview

A web base cost free platform where parents can find required tutor for their children & students can find their part time job like tuition. Nowadays we as university cannot find suitable tuition which helps us . Also there is so called media which created a lot of problems to find tuition. To overcome this problem Md Arham Ahmad Adil (Reg:**2019331005**) & I (Reg : **2019331075**) will work together in this project.

Goals

1. **To make a cost free platform:** There will be no cost during finding or registration in this platform . Anyone can search for a suitable job or parents can post a job.
2. **Fill up expectation:** To fill parents or students expectations as they require.
3. **Reduce dependency on media:** A new student won't be harmed by so called media.

Specifications

-User:Parents & Tutor

-Registration via require data

-Login via email & password

-Parents can post a new tuition, approve or reject any applicant and delete a responded tuition.

-Tutor can choose tuition, connect with parents by phone number & get parent’s response.

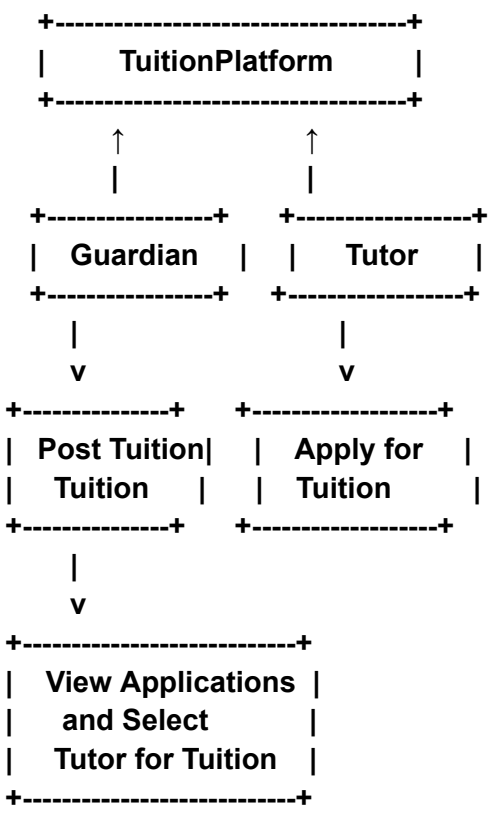
Milestones

1. Project Initiation and Planning

2. Platform Design

3. User Interface Refinement and Feature Development

Use Case Diagram



Actors:

Guardian: A user who wants to post tuition requirements and find tutors for their students.

Tutor: A user who wants to apply for a tutoring posted by guardians.

Use Cases:

Post Tuition:

A guardian creates a new tuition post by providing details like subject, grade level, and location.

Apply for Tuition:

A tutor applies for a tuition post.

A guardian views the list of tutor applications for their posted tuition.
The guardian selects a tutor from the list to conduct tutoring sessions.

Interactions:

A guardian interacts with the system to post tuition and view tutor applications.
A tutor interacts with the system to apply for tuition.
The system facilitates the matching of tutors with tuition posts and allows guardians to select tutors for their tuition.

Functional requirements

User Registration:

Users should be able to create accounts by providing their email, username, and password.
The system should validate the uniqueness of usernames and email addresses.

Login and Authentication:

Users should be able to log in using their registered email and password.
The system should authenticate users' credentials and grant access to authorized users.

Create and Post Tuition:

Guardians should be able to create tuition posts by providing details such as subject, grade level, location, and additional requirements.

Apply for Tuition:

Tutors should be able to apply for tuition posts.

View Applications:

Guardians should be able to view a list of tutor applications for each of their posted tuitions.

Search and Filtering:

Users should be able to search for available tuition based on criteria like subject.

Guardian Response:

A guardian can approve or reject any tutor who applied for the tuition. And the tutor can see response information in his inbox.

How to Run the Project

Software: xampp, apache server, MySQL

1. First we run apache server and MySQL from the xampp console locally.
2. Secondly we run "http://localhost/TuitionPlatform/index.php" in any browser.
3. Here, TuitionPlatform should be located inside the **xampp/htdocs** folder.

FrontEnd Development Tools: HTML, CSS, Bootstrap, JS

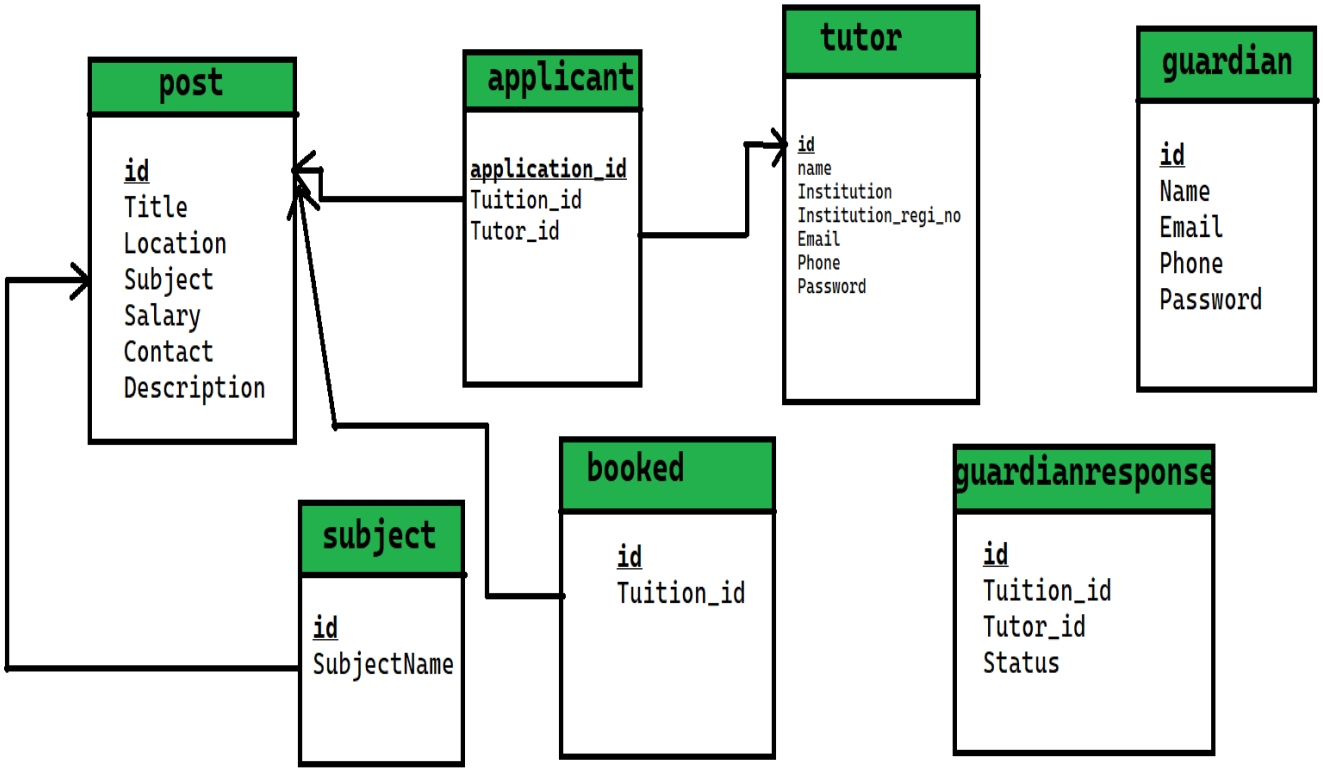
BackEnd Development Tools: PHP, MySQL Database

Data Definition Language

```
create table post(
  Title varchar(255);
  Location varchar(50),
  Subject varchar(20),
  Salary varchar(5),
  Contact char(11),
  Description varchar(255)
  primary key (id)
);
create table subject(
  SubjectName varchar(20) not null,
  int id,
  foreign key(id) references post(id)
  on delete cascade
);

create table guardian(
  name varchar(20),
  email varchar(20),
  phone char(11),
  password varchar(10)
);

create table tutor(
  name varchar(20),
  Institution varchar(20),
  Institution_regi_no varchar(20),
  email varchar(20),
  phone char(11),
  password varchar(10)
);
```

```
CREATE table applicant (
  application_id int,  Tuition_id int,
  Tutor_id INT,
  primary key(application_id),
  foreign key(Tuition_id) references post(id)
  on delete cascade,
  foreign key(Tutor_id) references tutor(id)
  on delete cascade
);

create guardianresponse (
  id INT,
  Tuition_id INT,
  Tutor_id, INT,
  Status VARCHAR(20)
);
```

Schema Diagram



Update

New Database Tables:

```
booked (
        id INT,
        Tuition_id INT
        foreign key(Tuition_id) references post(id)
        on delete cascade

);
```

Improvements to Project:

1. If any tutor applied for a tuition post then that post will be ‘Booked’.
2. If guardian wants to delete a tuition post then their will be an alert message to confirm Deletion.