Project Group 9 - ShareNLearn

Team Members:

Rahul Ratra **RaviKumar Mittapalli** **Sahithi Priya Gutta** **Noor Zahara**

PROJECT PROPOSAL

Introduction:

ShareNLearn is an online portal to help people find a tutor or group of people to learn/discuss any topic or subject they are having trouble understanding it. A search using **zip code** or **subject** helps one to find a tutor or people interested in group learning and discussion. Every person will be charged based on the usage of service (i.e. on number of tutors they are connected with, number of subjects they are getting help for etc). Tutors can also put up there proposal fee for each class they teach, which helps for student to select tutor based on their budget. Group discussion service will be charged for each person involved in the discussion.

Scope:

Anyone can register themselves as tutors or students(any user who is not tutor is considered as student) or can be both. They will have separate dashboard which shows their respective student schedules or teaching schedules. They can register with their personal details like name, email, phone, address and the type they are registering as i.e tutor, student or both. The amount will be charged only when the services are availed and not during registration. The tutor has an option to teach in-person, conduct an online class, upload videos of each of his/her class or take a live online class.

Student will be charged for each class and payment will be charged based on many factors. Website will charge tutors based on the amount paid by students(i.e A percentage of the amount will be paid for the website by the tutor as a service charge).Only after the payment is done they are allowed to talk to/contact the tutor/fellow group member.

Project Group 9 - ShareNLearn

Team Members:

Rahul Ratra **RaviKumar Mittapalli** **Sahithi Priya Gutta** **Noor Zahara**

Objectives:

Major objectives are:

- To implement interactive and user friendly interface where students can find tutors and take their help online or in person.
- An online platform where tutor can register and provide the services to registered students online or in person.(We are basically focussed on database activities which take care of backend business logic while charging students)
- Should implement a calendar system which gives clear schedules for each user(Which gives you info about free and busy times)
- Should implement payment systems which will automatically update the status in the DB for each class availability for students based on payment dues.

Project Group 9 - ShareNLearn

Team Members:

Rahul Ratra **RaviKumar Mittapalli** **Sahithi Priya Gutta** **Noor Zahara**

Sprint 0

Project Environment

The technologies that will be used : MySQL, Spring Boot framework, HTML, Javascript.
Programming Language : Advanced Java

High - Level Conceptual Design

User Roles

User Role	Description
Student	The Person who availed the services and take classes under tutor is considered student for that service
Tutor	The Person who offers the classes and teach people in the expertise subjects and topics is considered as tutor

User Stories :

Story ID	Description
US1	User can register as student, tutor or both.
US2	User as a student can search for nearby services.
US3	User as a student can pay for the services availed.
US4	User as a student can register for the available services.

Project Group 9 - ShareNLearn

Team Members:

Rahul Ratra **RaviKumar Mittapalli** **Sahithi Priya Gutta** **Noor Zahara**

US5	User as a tutor can add/modify his expertise for which he will provide the services.
US6	User as a tutor has to provide information about his preference about online or in person services.
US7	User as a tutor can add/modify his timings when he is going to add/modify the requested service.
US8	User can form a group for discussing on related topics.
US9	User can login as a student or tutor or both.
US10	User as a tutor can view the service details he is offering.
US11	User as tutor can modify the status of his/her services.
US12	User registered as both student and tutor has access to both student and tutor dashboards.

Entities:

User
User_role
Category
Subject
Topic
Services_types
Tutor_schedule

Entity Relationships:

Project Group 9 - ShareNLearn

Team Members:

Rahul Ratra **RaviKumar Mittapalli** **Sahithi Priya Gutta** **Noor Zahara**

1. User register for a Services_types.
Cardinality: Many to Many
Participation: User has partial participation.
Services_types has partial participation.

2. User adds/modifies his Tutor_schedule.
Cardinality: One to Many
Participation: Tutor_schedule has total participation.
User has partial participation.

3. Category has Subjects.
Cardinality: One to Many
Participation: Category has total participation.
Subjects has total participation.

4. Subject has Topics.
Cardinality: One to Many
Participation: Subject has total participation.
Topics has total participation.

5. User has a User role.
Cardinality: Many to Many
Participation: User has total participation.
User role has total participation.

6. User can avail the Services availed.
Cardinality: Many to Many
Participation: User has partial participation.
Services availed has partial participation.

7. User can activate/de-activate the Services availed.
Cardinality: One to Many
Participation: User has partial participation.
Services has partial participation.

8. User as tutor teaches Topics
Cardinality: Many to Many
Participation: User has partial participation.
Topics has partial participation.

9. User as a student learn Topics
Cardinality: Many to Many
Participation: User has partial participation.

Project Group 9 - ShareNLearn

Team Members:

Rahul Ratra **RaviKumar Mittapalli** **Sahithi Priya Gutta** **Noor Zahara**

Topics has partial participation.

Sprint 1

REQUIREMENTS

User Stories considered for this Sprint:

Story ID	Description
US1	User can register as student, tutor or both.
US2	User as a student can search for nearby services.
US3	User as a student can pay for the services availed.
US4	User as a student can register for the available services.

CONCEPTUAL DESIGN

Entity: User

Project Group 9 - ShareNLearn

Team Members:

Rahul Ratra **RaviKumar Mittapalli** **Sahithi Priya Gutta** **Noor Zahara**

Attributes:

- SSN
- Name [Composite]
  - First_Name
  - Middle_Name
  - Last_Name
- Address [Composite]
  - Address_Line_1
  - Address_Line_2
- City
- State
- Country
- Zip_Code

Entity: User_Role

Attributes:

- SSN
- User_Type [Single-Valued]

Entity: Category

Attributes:

- Category_Name [Single-Valued]

Entity: Subject

Attributes:

- Subject_Name [Single-Valued]

Entity: Topic

Attributes:

- Topic_Name [Single-Valued]

Entity: Services_Types

Attributes:

- Service_Type [Single-Valued]

Project Group 9 - ShareNLearn

Team Members:
Rahul Ratra **RaviKumar Mittapalli** **Sahithi Priya Gutta** **Noor Zahara**

Entity: Tutor_Schedule
Attributes:

User_SSN
Topic_Name
Class_Size [Single_Valued]

Entity Relationships:

1. User as tutor register for a Services_types.
Cardinality: Many to Many
Participation: User has partial participation.
Services_types has partial participation.

2. User as tutor adds/modifies his Tutor_schedule.
Cardinality: One to Many
Participation: Tutor_schedule has total participation.
User has partial participation.

3. Category has Subjects.
Cardinality: One to Many
Participation: Category has total participation.
Subjects has total participation.

4. Subject has Topics.
Cardinality: One to Many
Participation: Subject has total participation.
Topics has total participation.

5. User has a User role.
Cardinality: Many to Many
Participation: User has total participation.
User role has total participation.

6. User can avail the Services_Types.
Cardinality: Many to Many

Participation: User has partial participation.
Services availed has partial participation.

7. User as tutor teaches Topics
Cardinality: Many to Many
Participation: User has partial participation.
Topics has partial participation.

8. User as a student learn Topics
Cardinality: Many to Many
Participation: User has partial participation.
Topics has partial participation.

LOGICAL DESIGN

Tables

Table 1: Users

Columns:

User_ID [Primary Key]
First_Name
Middle_Name
Last_Name
Email_Id
Address_Line1
Address_Line2
City
State
Country
Zip_Code

Primary Key Justification: User_ID is an auto-increment id which is unique, hence chosen as primary key.

Table 2: User_Role

Project Group 9 - ShareNLearn

Team Members:
Rahul Ratra **RaviKumar Mittapalli** **Sahithi Priya Gutta** **Noor Zahara**

Columns:
User_ID [Primary Key] [Foreign Key; references to table User]
User_Type

Primary Key Justification: User_ID is a primary key of User Table which is unique and hence can be used as primary for this table as well.

Table 3: Category
Columns:
Category_ID [Primary Key]
Category_Type

Primary Key Justification: Category_ID is an auto-increment id which is unique, hence chosen as primary key.

Table 4: Subjects
Columns:
Subject_ID [Primary Key]
Category_ID [Foreign Key; references to table Category]
Subject_Name

Primary Key Justification: Subject_ID is an auto-increment id which is unique, hence chosen as primary key.

Table 5: Topics
Columns:
Topic_ID [Primary Key]
Subject_ID [Foreign Key; references to table Subject]
Topic_Name

Primary Key Justification: Topic_ID is an auto-increment id which is unique, hence chosen as primary key.

Project Group 9 - ShareNLearn

Team Members:
Rahul Ratra **RaviKumar Mittapalli** **Sahithi Priya Gutta** **Noor Zahara**

Table 6: Services_Types
Columns:
Service_Type_ID [Primary Key]
Service_Type

Primary Key Justification: Service_Type_ID is an auto-increment id which is unique, hence chosen as primary key.

Table 7: Tutor_Schedule
Columns:
Schedule_ID [Primary Key]
User_ID [Foreign Key; references to table User]
Topic_ID [Foreign Key; references to table Topic]
Class_Size

Primary Key Justification: Schedule-ID is an auto-increment id which is unique, hence chosen as primary key.

Table 8: Availability Timings
Columns:
Tutor_Availablility_ID [Primary Key]
Schedule_ID [Foreign Key; references to table Tutor_Schedule]
Service_Type_ID [Foreign Key; references to table Services_Types]
Week_Day
Avail_Time_From
Avail_Time_To

Primary Key Justification: Tutor_Availablility_ID is an auto-increment id which is unique, hence chosen as primary key.

Table 9: Services_Available
Columns:
Service_ID [Primary Key]

Team Members:
Rahul Ratra **RaviKumar Mittapalli** **Sahithi Priya Gutta** **Noor Zahara**

User_ID [Foreign Key; references to table User]
Topic_ID [Foreign Key; references to table Topic]
Schedule_ID [Foreign Key; references to table Tutor_Schedule]
Service_Status

Primary Key Justification: Service_ID is an auto-increment id which is unique, hence chosen as primary key.

Table 10: Student_Services
Columns:
User_ID [Primary Key] [Foreign Key; references to table User]
Service_ID [Primary Key] [Foreign Key; references to table Services_Available]
Paid_Status
Availabilty_Status

Primary Key Justi
fication: User_ID is a primary key of User Table which is unique and hence can be used as primary for this table as well.

SQL QUERIES:

1. **Search Query:**

```
select usr.first_name, usr.last_name,sb.subject_name,sb.subject_id,
tp.topic_id
 from services_available sa
join users usr on sa.user_id = usr.user_id
inner join user_role ur on ur.user_id = usr.user_id and user_type = 'T'
join tutor_schedule ts on ts.user_id = usr.user_id
#join avialability_timigs at on at.schedule_id = ts.schedule_id
join topics tp on tp.topic_id = ts.topic_id
join subjects sb on sb.subject_id = tp.subject_id
 where usr.zipcode = ? or sb.subject_name like ?
 or (usr.zipcode = ? and sb.subject_name like ?);
```

Project Group 9 - ShareNLearn

Team Members:

Rahul Ratra **RaviKumar Mittapalli** **Sahithi Priya Gutta** **Noor Zahara**

2. Register Query:

```
insert into users (first_name, middle_name, last_name,
email_id,address_line_1,address_line_2,city,state,country,zipcode)
values (?,?,?,?,?,?,?,?,?,?,?);

insert into user_role(user_id,user_type) values( ?,?);
```

3. Tutor Availability:

```
select * from avialability_timigs where schedule_id = ?;
```

4. Pay for services

```
update student_services set paid_status = 'PAID' where user_id = ? and service_id = ?
```

5. Register for services

```
insert into student_services (service_id,paid_status,availability_status)
values (?,'NOT_PAID',0);
```

Project Group 9 - ShareNLearn

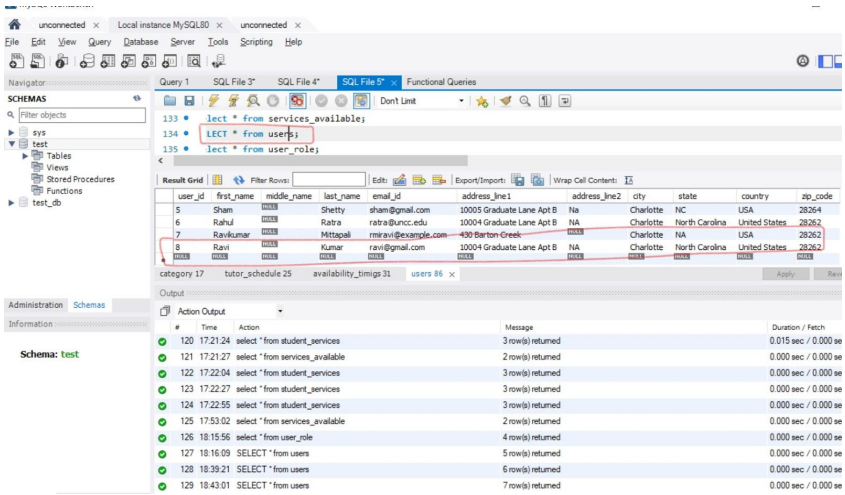
Team Members:

Rahul Ratra **RaviKumar Mittapalli** **Sahithi Priya Gutta** **Noor Zahara**

UI Demo :

1. User Registration

Entry in users table

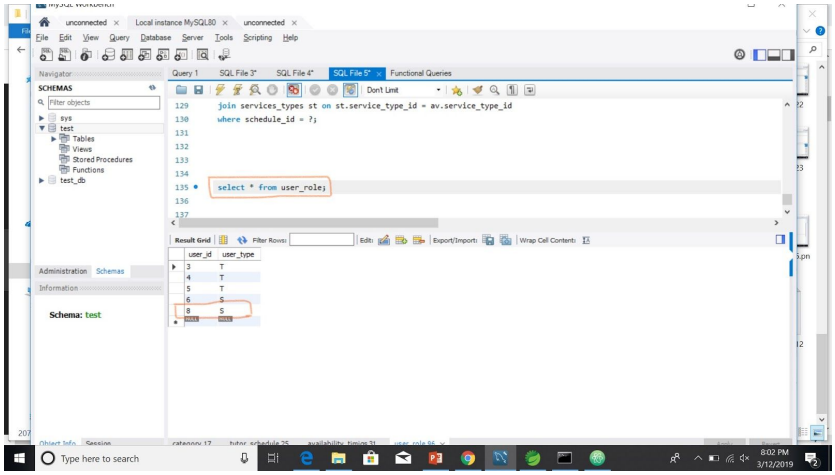


Project Group 9 - ShareNLearn

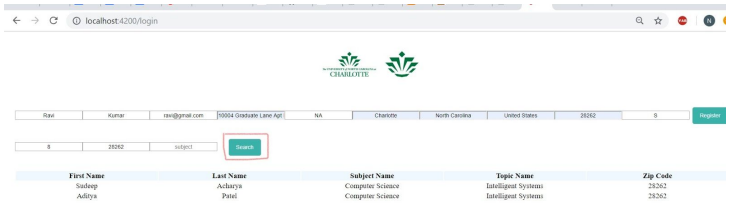
Team Members:

Rahul Ratra **RaviKumar Mittapalli** **Sahithi Priya Gutta** **Noor Zahara**

Entry in User_Role Table



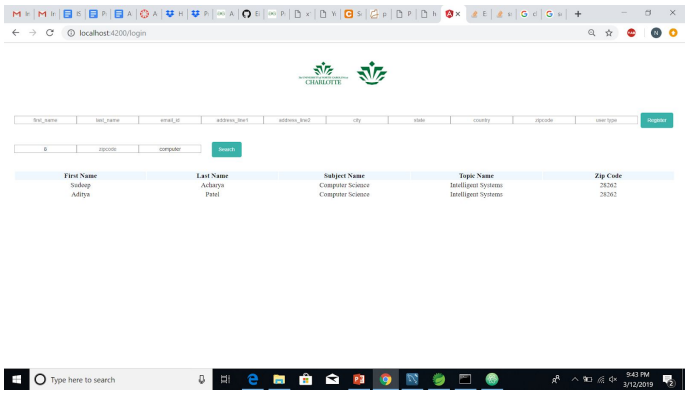
2. User search for available courses



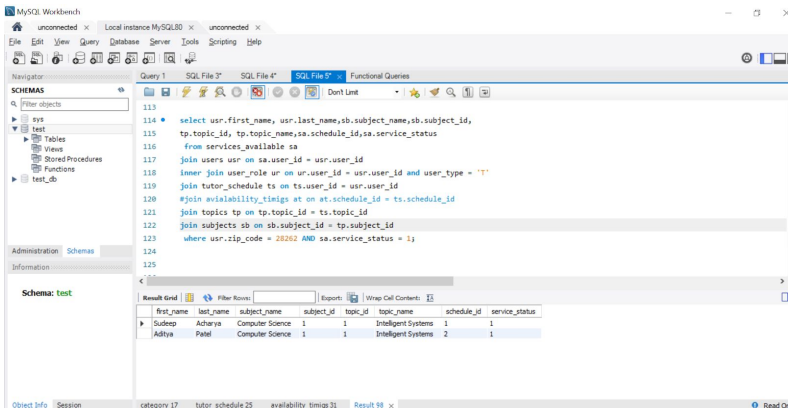
Project Group 9 - ShareNLearn

Team Members:

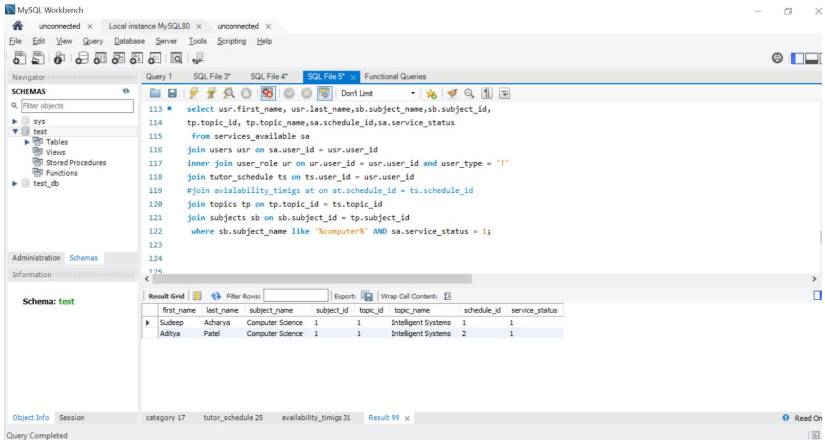
Rahul Ratra **RaviKumar Mittapalli** **Sahithi Priya Gutta** **Noor Zahara**



Query for available services searching using zip code



Query for available services searching using subject name

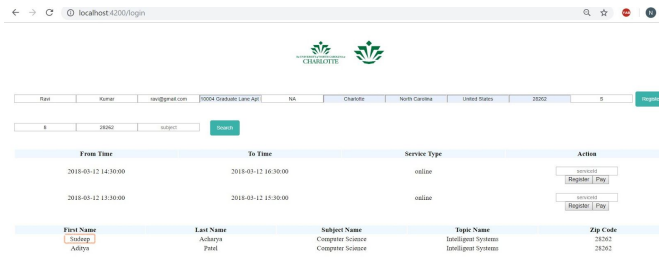


Tutor Availability Details

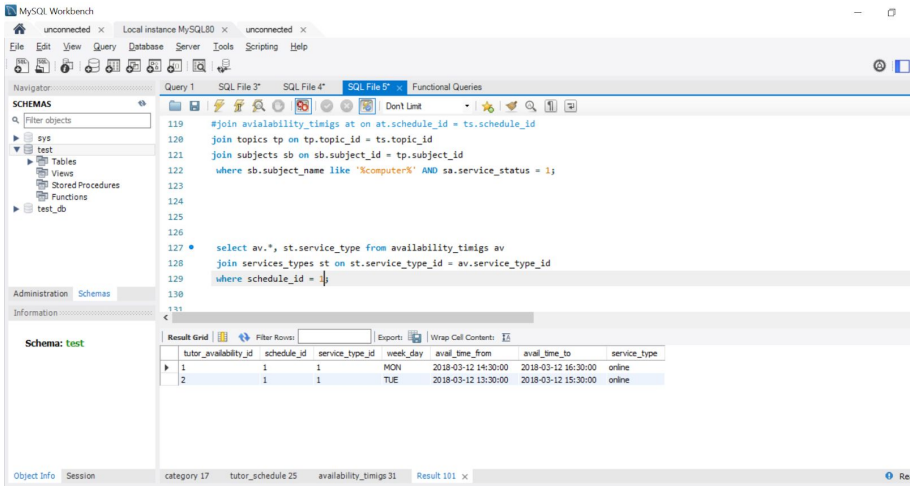
Project Group 9 - ShareNLearn

Team Members:

Rahul Ratra **RaviKumar Mittapalli** **Sahithi Priya Gutta** **Noor Zahara**



Query used for Fetching tutor availability

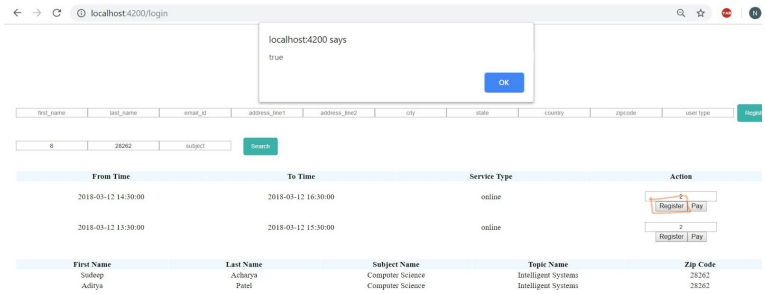


3. Register for available services

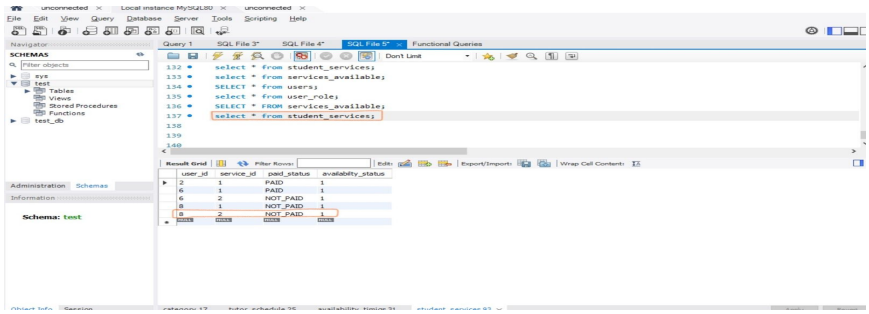
Project Group 9 - ShareNLearn

Team Members:

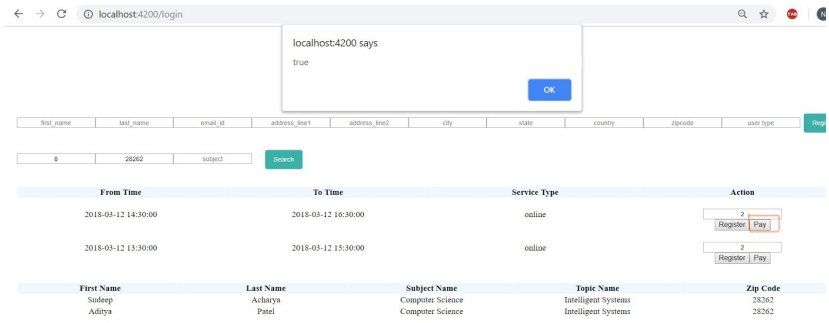
Rahul Ratra **RaviKumar Mittapalli** **Sahithi Priya Gutta** **Noor Zahara**



DB Entry post registration



4. User pays for the services availed

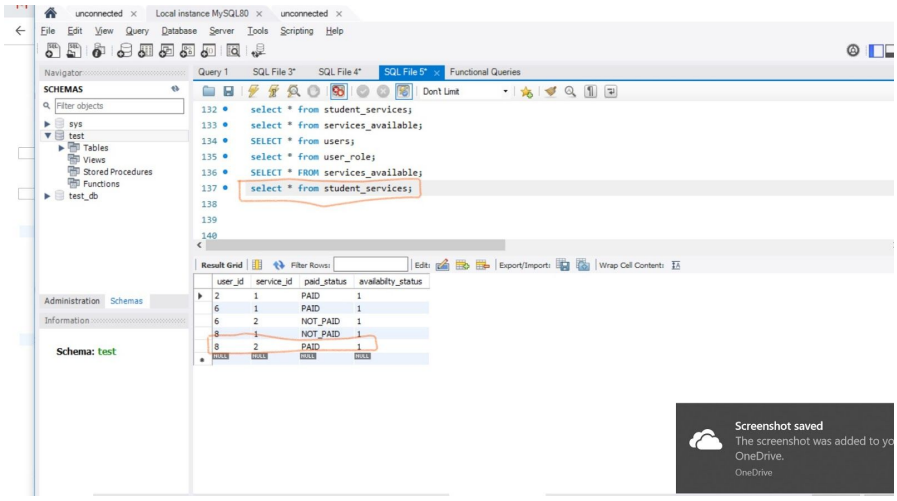


DB Update post payment

Project Group 9 - ShareNLearn

Team Members:

Rahul Ratra **RaviKumar Mittapalli** **Sahithi Priya Gutta** **Noor Zahara**



Sprint 2

REQUIREMENTS

User Stories considered for this Sprint:

Story ID	Description
US1	User can register as student, tutor or both.
US2	User as a student can search for nearby services.
US3	User as a student can pay for the services availed.
US4	User as a student can register for the available services.
US5	User can login as a student or tutor or both.
US6	User as a tutor can view the service details he is offering.

Project Group 9 - ShareNLearn

Team Members:

Rahul Ratra **RaviKumar Mittapalli** **Sahithi Priya Gutta** **Noor Zahara**

US7	User as tutor can modify the status of his/her services.
US8	User registered as both student and tutor has access to both student and tutor dashboards.

Entities:
User
Student_Role
Tutor_Role
Category
Subject
Topic
Services_types
Tutor_schedule

CONCEPTUAL DESIGN

Entity: User
Attributes:

- SSN
- Name [Composite]
  - First_Name
  - Middle_Name
  - Last_Name
- Email [Single-Valued]
- Password
- Address [Composite]
  - Address_Line_1
  - Address_Line_2
- City
- State

Project Group 9 - ShareNLearn

Team Members:

Rahul Ratra **RaviKumar Mittapalli** **Sahithi Priya Gutta** **Noor Zahara**

Country
Zip_Code

Entity: Student_Role
Attributes:
SSN

Entity: Tutor_Role
Attributes:
SSN

Entity: Category
Attributes:
Category_Name [Single-Valued]

Entity: Subject
Attributes:
Subject_Name [Single-Valued]

Entity: Topic
Attributes:
Topic_Name [Single-Valued]

Entity: Services_Types
Attributes:
Service_Type [Single-Valued]

Entity: Tutor_Schedule
Attributes:
User_SSN
Topic_Name
Class_Size [Single_Valued]

Project Group 9 - ShareNLearn

Team Members:

Rahul Ratra **RaviKumar Mittapalli** **Sahithi Priya Gutta** **Noor Zahara**

Entity Relationships:

1. User as tutor can register for a Services_types.
Cardinality: Many to Many
Participation: User has partial participation.
Services_types has partial participation.
2. User as tutor adds/modifies his Tutor_schedule.
Cardinality: One to Many
Participation: Tutor_schedule has total participation.
User has partial participation.
3. Category has Subjects.
Cardinality: One to Many
Participation: Category has total participation.
Subjects has total participation.
4. Subject has Topics.
Cardinality: One to Many
Participation: Subject has total participation.
Topics has total participation.
5. User has a Student/Tutor role.
Cardinality: Many to One
Participation: User has total participation.
Student/Tutor role has partial participation.
6. User can avail the Services_Types.
Cardinality: Many to Many
Participation: User has partial participation.
Services availed has partial participation.
7. User as tutor teaches Topics.
Cardinality: Many to Many
Participation: User has partial participation.
Topics has partial participation.
8. User as a student learn Topics.
Cardinality: Many to Many
Participation: User has partial participation.
Topics has partial participation.
9. User can login as a student/tutor .

Project Group 9 - ShareNLearn

Team Members:

Rahul Ratra **RaviKumar Mittapalli** **Sahithi Priya Gutta** **Noor Zahara**

Cardinality: Many to One

Participation: User has total participation.

Student_Role/Tutor_Role has partial participation.

LOGICAL DESIGN WITH NORMAL FORM IDENTIFICATION

Tables

Table 1: Users

Columns:

User_ID [Primary Key]
First_Name
Middle_Name
Last_Name
Email_Id
Password [Unique_Key]
Address_Line1
Address_Line2
City
State
Country
Zip_Code

Primary Key Justification: User_ID is an auto-increment id which is unique, hence chosen as primary key.

Highest Normalization Level: It is in 2NF.

Justification: It is in 2NF as state,city depends on zip code. We don't need to save address separately so we just saved it in the same table i.e, user.

Table 2: Student_Role
Columns:
User_ID [Primary Key] [Foreign Key; references to table User]

Primary Key Justification: User_ID is a primary key of User Table which is unique and hence can be used as primary for this table as well.
Highest Normalization Level: It is in 4NF.

Table 3: Tutor_Role
Columns:
User_ID [Primary Key] [Foreign Key; references to table User]

Primary Key Justification: User_ID is a primary key of User Table which is unique and hence can be used as primary for this table as well.
Highest Normalization Level: It is in 4NF.

Table 4: Category
Columns:
Category_ID [Primary Key]
Category_Type

Primary Key Justification: Category_ID is an auto-increment id which is unique, hence chosen as primary key.
Highest Normalization Level: It is in 4NF.

Table 5: Subjects
Columns:
Subject_ID [Primary Key]
Category_ID [Foreign Key; references to table Category]

Project Group 9 - ShareNLearn

Team Members:
Rahul Ratra **RaviKumar Mittapalli** **Sahithi Priya Gutta** **Noor Zahara**

Subject_Name

Primary Key Justification: Subject_ID is an auto-increment id which is unique, hence chosen as primary key.
Highest Normalization Level: It is in 4NF.

Table 6: Topics
Columns:
Topic_ID [Primary Key]
Subject_ID [Foreign Key; references to table Subject]
Topic_Name

Primary Key Justification: Topic_ID is an auto-increment id which is unique, hence chosen as primary key.
Highest Normalization Level: It is in 4NF.

Table 7: Services_Types
Columns:
Service_Type_ID [Primary Key]
Service_Type

Primary Key Justification: Service_Type_ID is an auto-increment id which is unique, hence chosen as primary key.
Highest Normalization Level: It is in 4NF.

Table 8: Tutor_Schedule
Columns:
Schedule_ID [Primary Key]
User_ID [Foreign Key; references to table Tutor_Role]
Topic_ID [Foreign Key; references to table Topic]
Class_Size

Primary Key Justification: Schedule-ID is an auto-increment id which is unique, hence chosen as primary key.
Highest Normalization Level: It is in 4NF.

Table 9: Availability Timings
Columns:
Tutor_Availablility_ID [Primary Key]
Schedule_ID [Foreign Key; references to table Tutor_Schedule]
Service_Type_ID [Foreign Key; references to table Services_Types]
Week_Day
Avail_Time_From
Avail_Time_To

Primary Key Justification: Tutor_Availablility_ID is an auto-increment id which is unique, hence chosen as primary key.
Highest Normalization Level: It is in 4NF.

Table 10: Services_Available
Columns:
Service_ID [Primary Key]
User_ID [Foreign Key; references to table Tutor_Role]
Topic_ID [Foreign Key; references to table Topic]
Schedule_ID [Foreign Key; references to table Tutor_Schedule]
Service_Status

Primary Key Justification: Service_ID is an auto-increment id which is unique, hence chosen as primary key.
Highest Normalization Level: It is in 4NF.

Table 11: Student_Services
Columns:
User_ID [Primary Key] [Foreign Key; references to table Student_Role]
Service_ID [Primary Key] [Foreign Key; references to table Services_Available]

Project Group 9 - ShareNLearn

Team Members:

Rahul Ratra **RaviKumar Mittapalli** **Sahithi Priya Gutta** **Noor Zahara**

Paid_Status

Availabilty_Status

Primary Key Justification: User_ID is a primary key of User Table which is unique and hence can be used as primary for this table as well.

Highest Normalization Level: It is in 4NF.

SQL QUERIES

1. **Registering Users:** insert into users (first_name, last_name,email_id, password,address_line1,address_line2,city,state,country,zip_code) values (?,?,?,?,?,?,?,?,?,?);

2. **Register as a Student:** insert into tutor_role (user_id) values(?)

3. **Register as a Tutor:** insert into student_role (user_id) values(?)

4. **Search for Available services using zip code/ subject_name:**

```
select usr.first_name, usr.last_name,sb.subject_name,sb.subject_id,
 tp.topic_id, tp.topic_name,sa.tutor_availability_id,usr.zip_code,sa.service_id
from services_available sa join users usr on sa.user_id = usr.user_id
inner join tutor_role ur on ur.user_id = usr.user_id
 join tutor_schedule ts on ts.user_id = usr.user_id  join topics tp on tp.topic_id = ts.topic_id
join subjects sb on sb.subject_id = tp.subject_id  where sa.service_status = 1 and  (usr.zip_code =
? OR sb.subject_name like ?)
```

Project Group 9 - ShareNLearn

Team Members:

Rahul Ratra **RaviKumar Mittapalli** **Sahithi Priya Gutta** **Noor Zahara**

5. **Get tutor schedule for the given tutor:**

```
select av.*, st.service_type,CASE WHEN (ss.user_id is not null and ss.paid_status = 'NOT_PAID') THEN 1 WHEN (ss.user_id is not null and ss.paid_status = 'PAID') THEN 2 ELSE 0 END registerFlag from availability_timigs av
join services_available sa on sa.tutor_availability_id = av.tutor_availability_id
 left join student_services ss on ss.service_id = sa.service_id and ss.user_id = ?
join services_types st on st.service_type_id = av.service_type_id where av.tutor_availability_id = ?
```

6. **Register for services:** insert into student_services (user_id,service_id,paid_status) values (?,?,'NOT_PAID')

7. **Pay for services:**

```
update student_services set paid_status = 'PAID',availabilty_status = 1 where user_id = ? and service_id = ?
```

8. **Login as Tutor/Student:**

```
select ifnull(tr.user_id,0) tutor, ifnull(sr.user_id,0) student from users usr
left join student_role sr on sr.user_id = usr.user_id
left join tutor_role tr on tr.user_id = usr.user_id where usr.user_id = ? and usr.password = ?
```

9. **Fetch Tutor Service Details:**

```
select topic_name, subject_name,category_type,avail_time_from, avail_time_to, service_type, class_size,at.tutor_availability_id,sa.service_status,sa.service_id from tutor_schedule ts join topics t on t.topic_id = ts.topic_id
join subjects s on s.subject_id = t.subject_id join category c on c.category_id = s.category_id
join availability_timigs at on at.schedule_id = ts.schedule_id join services_types st on st.service_type_id = at.service_type_id join services_available sa on sa.user_id = ts.user_id
and at.tutor_availability_id = sa.tutor_availability_id where ts.user_id  = ?
```

Project Group 9 - ShareNLearn

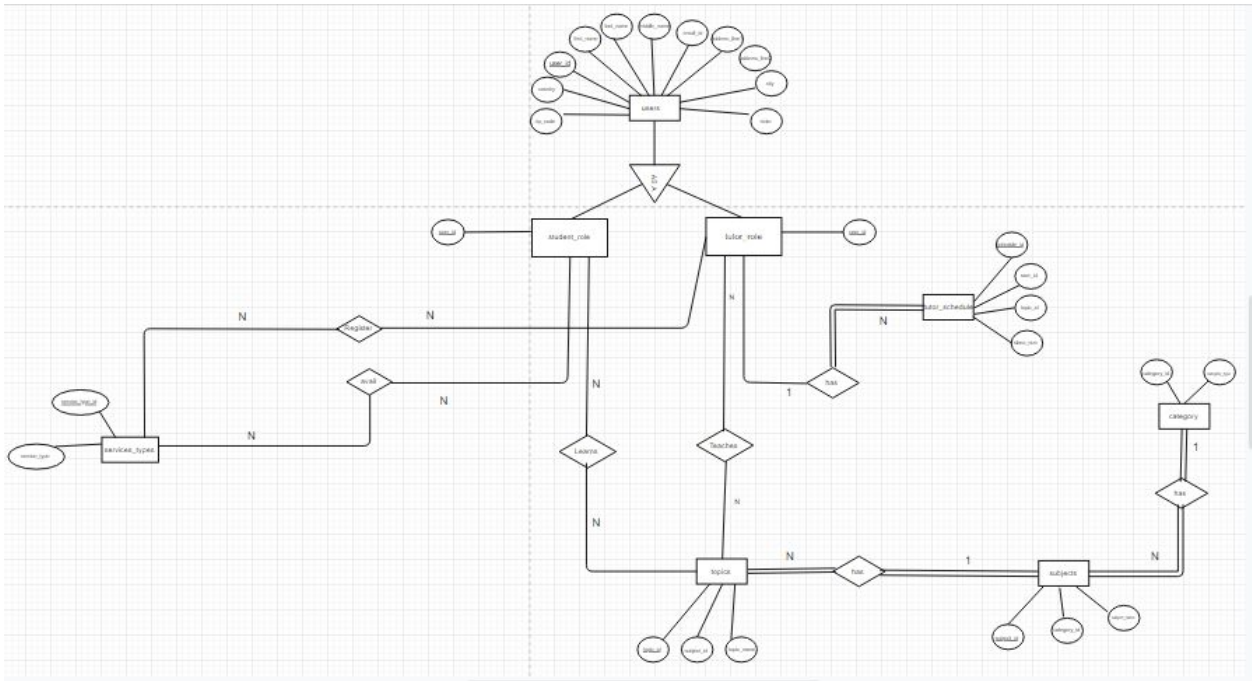
Team Members:

Rahul Ratra **RaviKumar Mittapalli** **Sahithi Priya Gutta** **Noor Zahara**

10. **Modify the status of services:**

update services_available set service_status = ? where service_id = ?

E-R Diagram



Project Group 9 - ShareNLearn

Team Members:

Rahul Ratra **RaviKumar Mittapalli** **Sahithi Priya Gutta** **Noor Zahara**

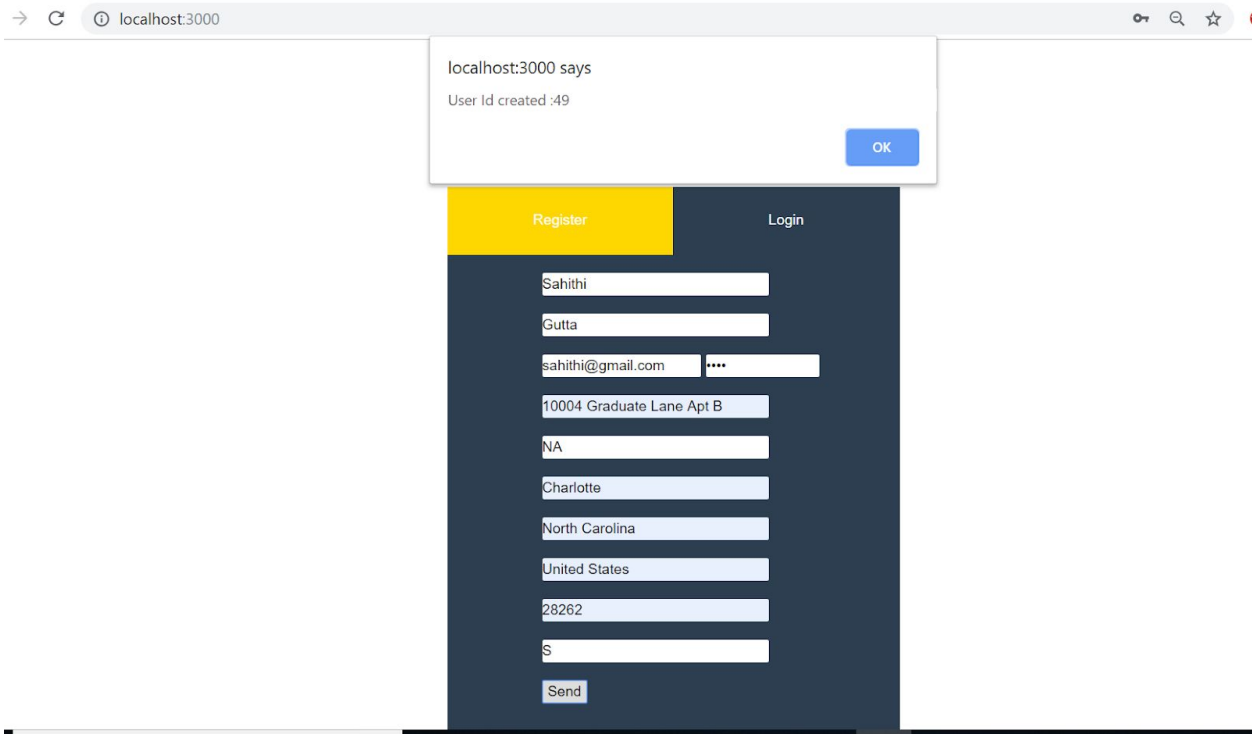
UI Demo

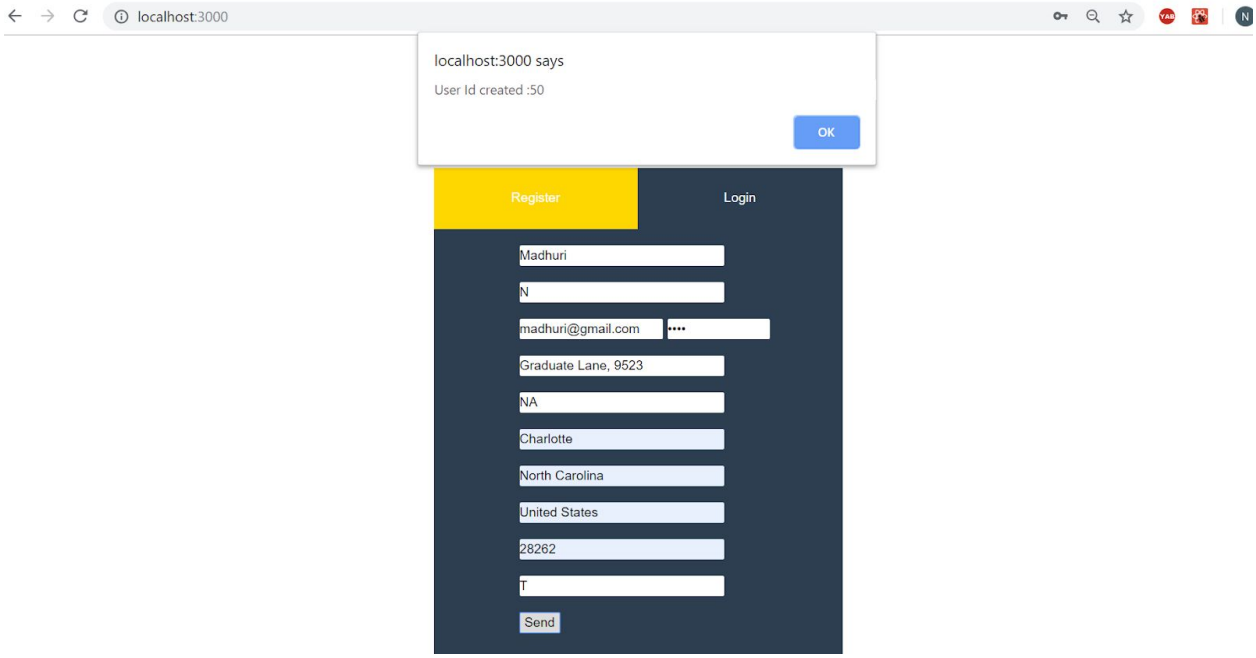
1. **Registration Feature**

Project Group 9 - ShareNLearn

Team Members:

Rahul Ratra **RaviKumar Mittapalli** **Sahithi Priya Gutta** **Noor Zahara**

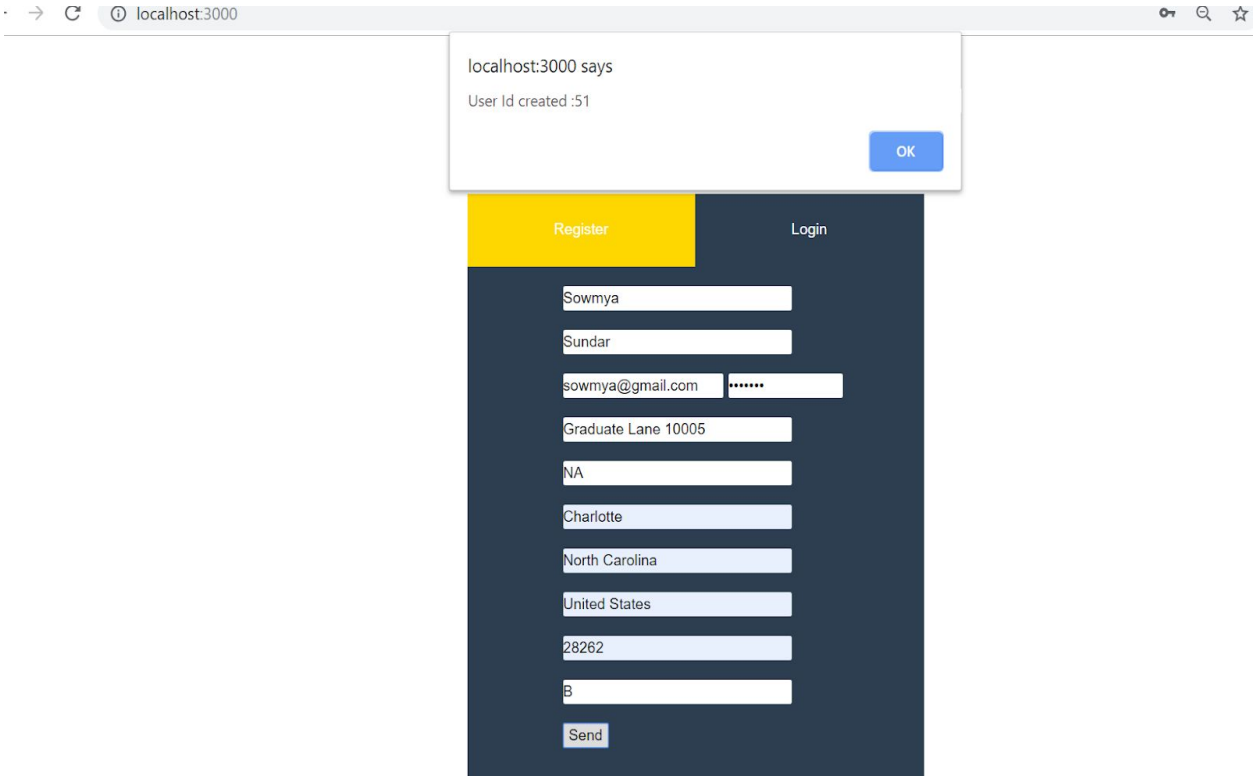


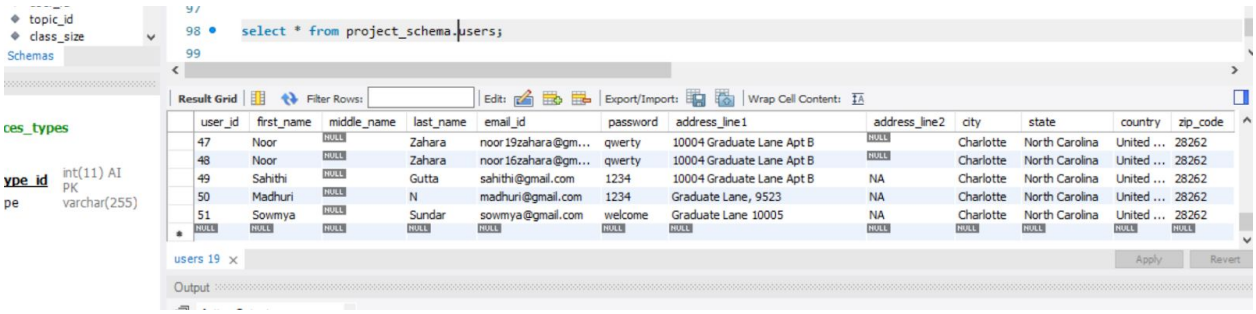


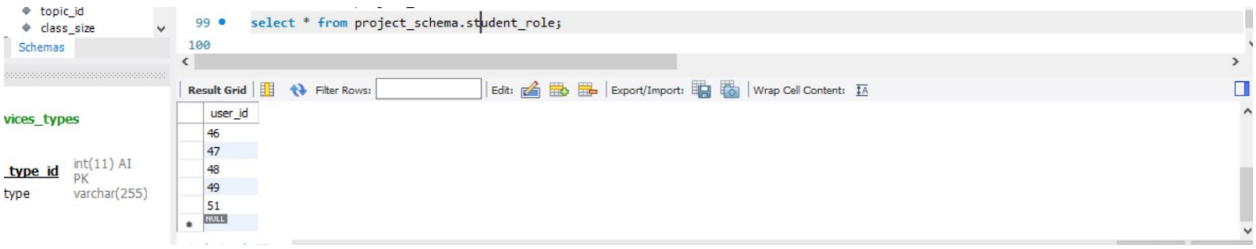
Project Group 9 - ShareNLearn

Team Members:

Rahul Ratra **RaviKumar Mittapalli** **Sahithi Priya Gutta** **Noor Zahara**



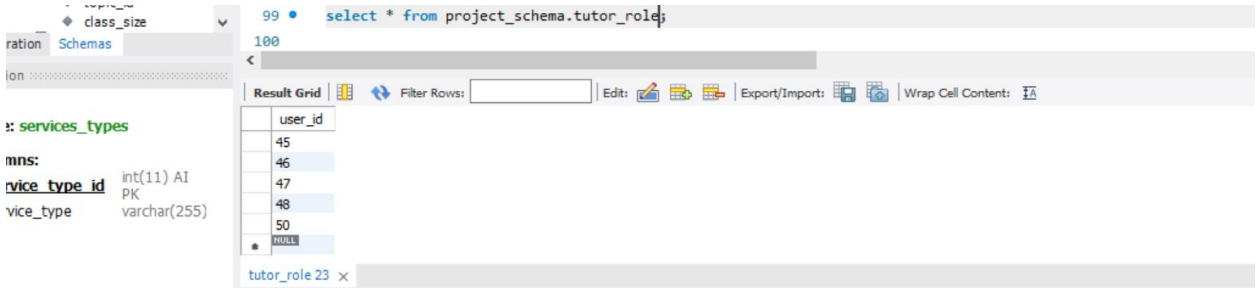




Project Group 9 - ShareNLearn

Team Members:

Rahul Ratra **RaviKumar Mittapalli** **Sahithi Priya Gutta** **Noor Zahara**



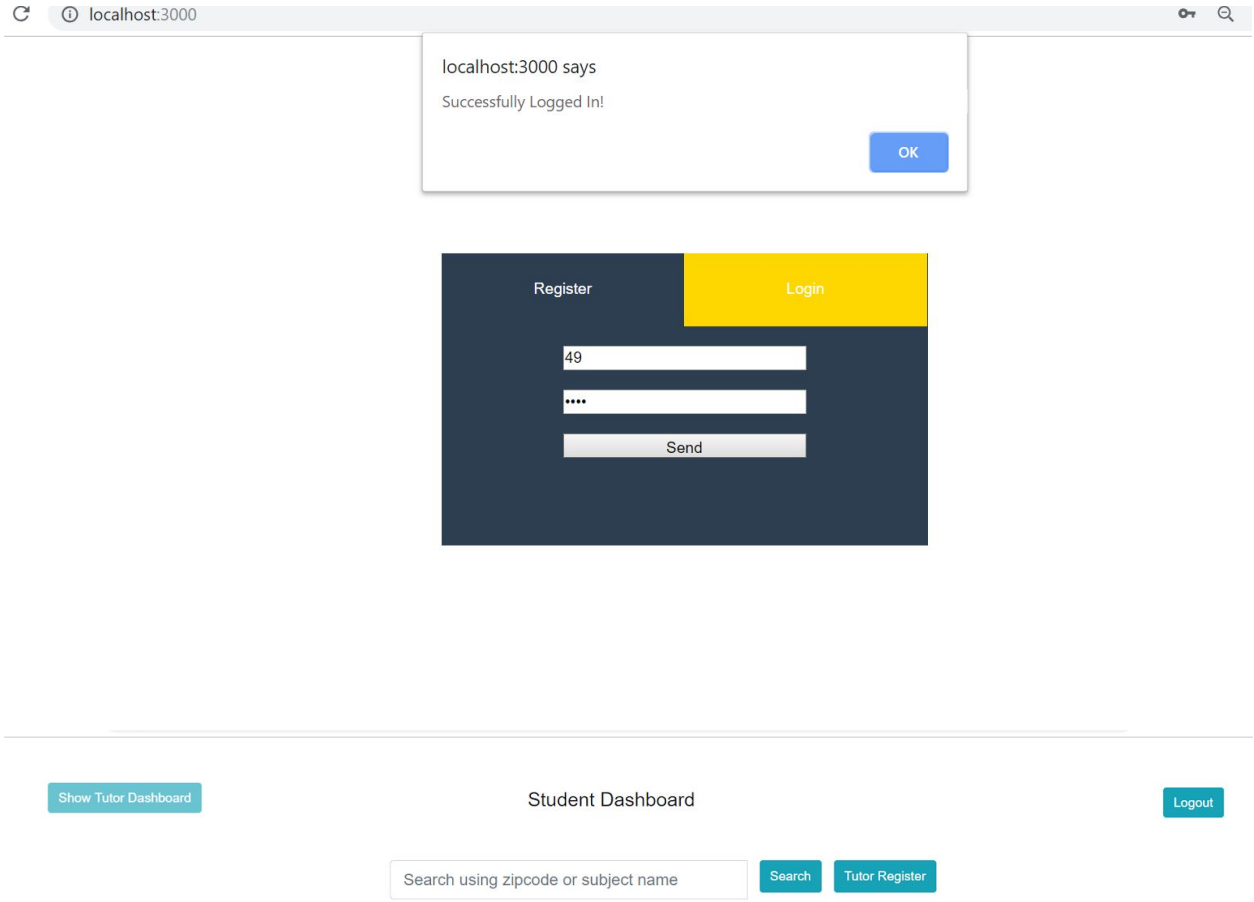
2. Login Feature

   a. As a student

Project Group 9 - ShareNLearn

Team Members:

Rahul Ratra **RaviKumar Mittapalli** **Sahithi Priya Gutta** **Noor Zahara**

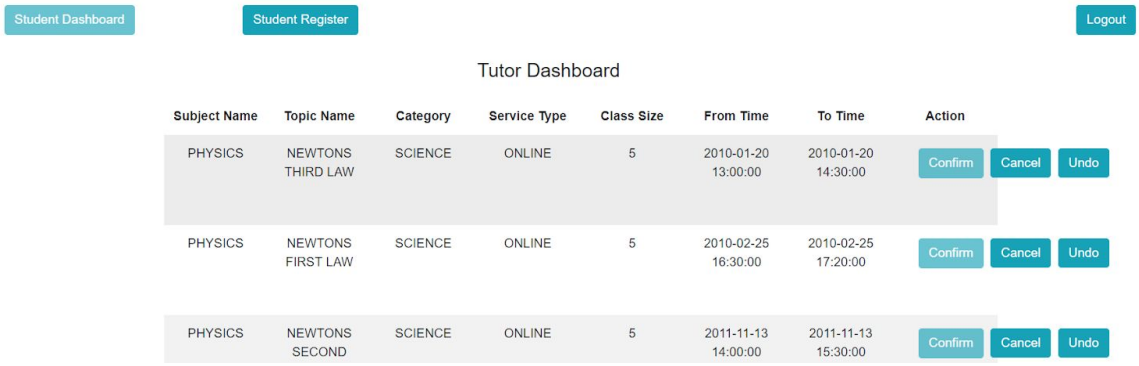


b. As a Tutor

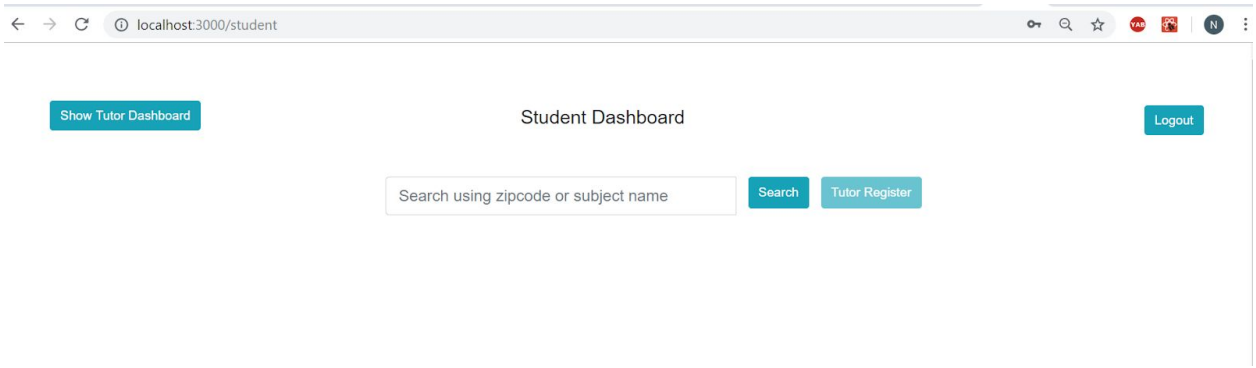
Project Group 9 - ShareNLearn

Team Members:

Rahul Ratra **RaviKumar Mittapalli** **Sahithi Priya Gutta** **Noor Zahara**



c. As Both

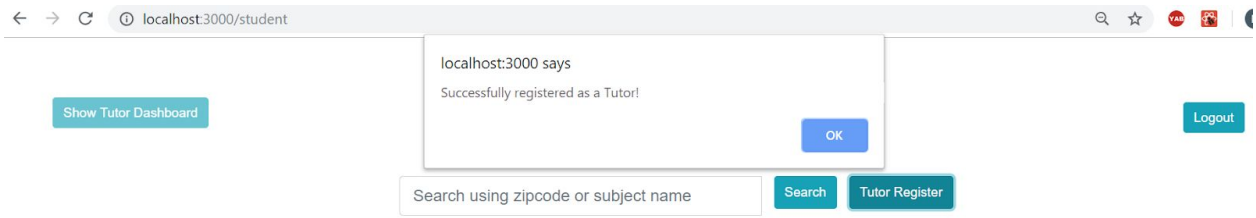


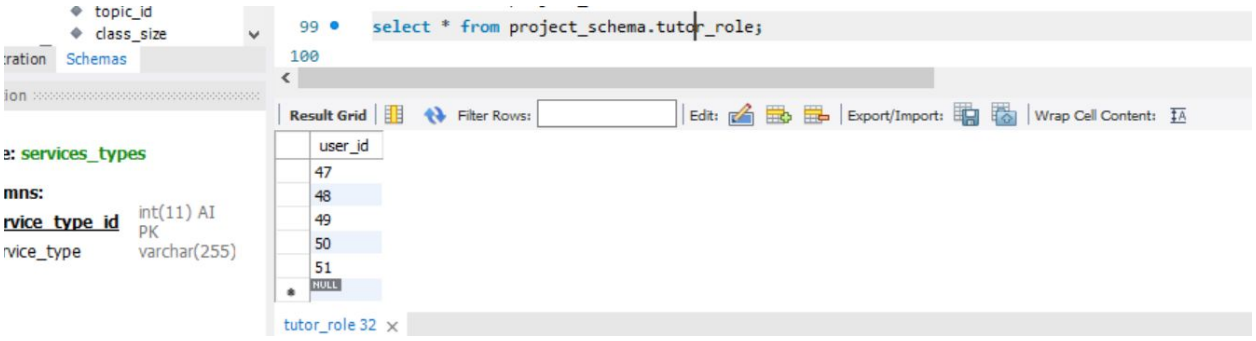
3. User as a student registers as a Tutor

Project Group 9 - ShareNLearn

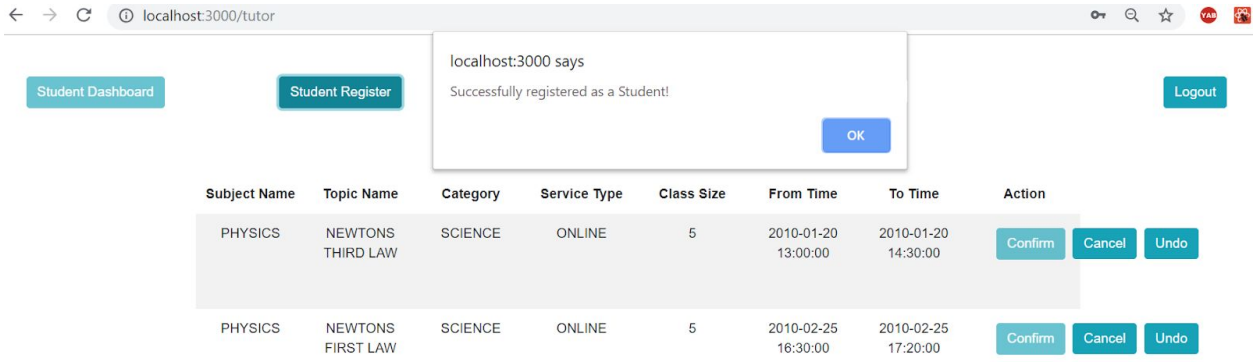
Team Members:

Rahul Ratra **RaviKumar Mittapalli** **Sahithi Priya Gutta** **Noor Zahara**





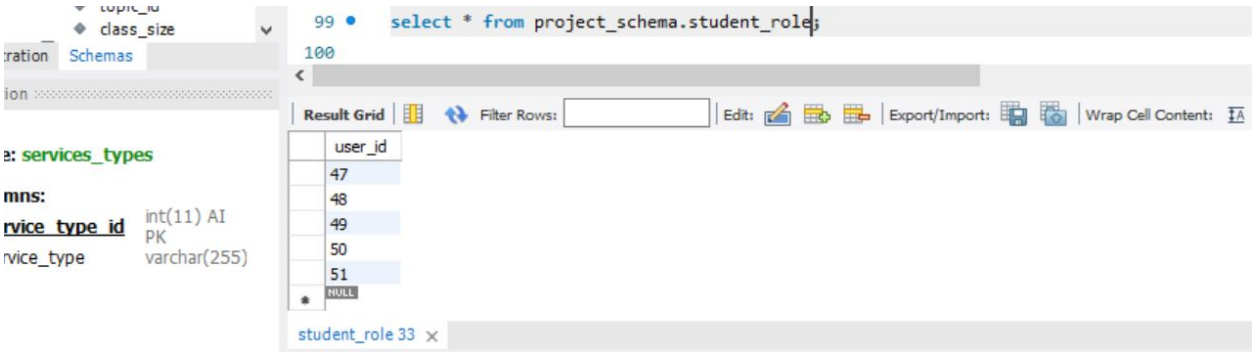
4. User as a Tutor registers as a Student



Project Group 9 - ShareNLearn

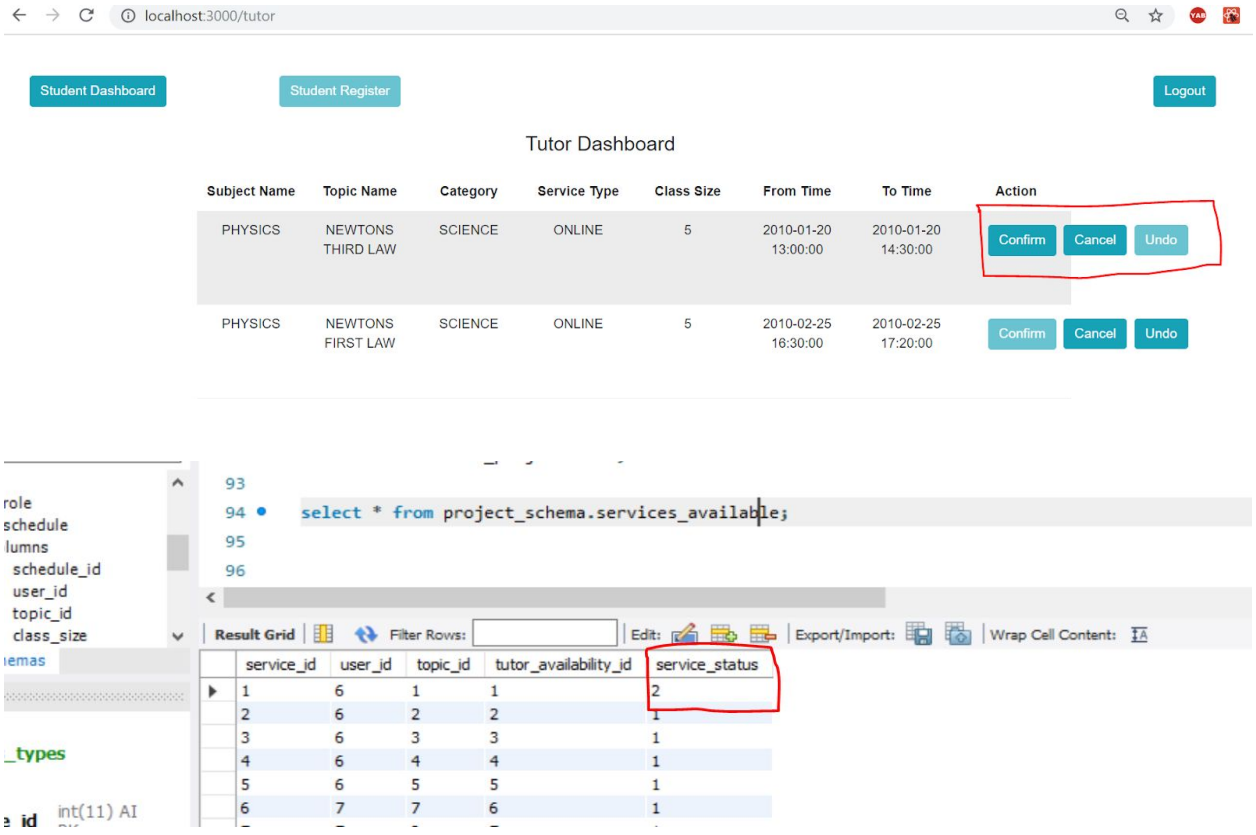
Team Members:

Rahul Ratra **RaviKumar Mittapalli** **Sahithi Priya Gutta** **Noor Zahara**



5. When service is in not confirmed state

service_status = 2



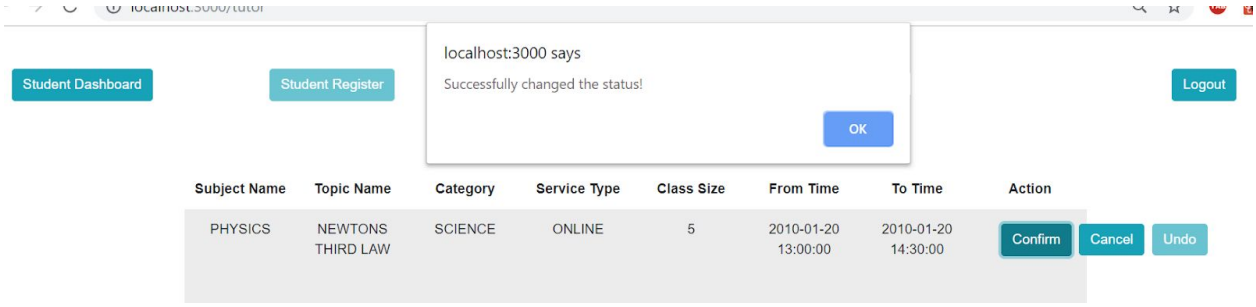
Project Group 9 - ShareNLearn

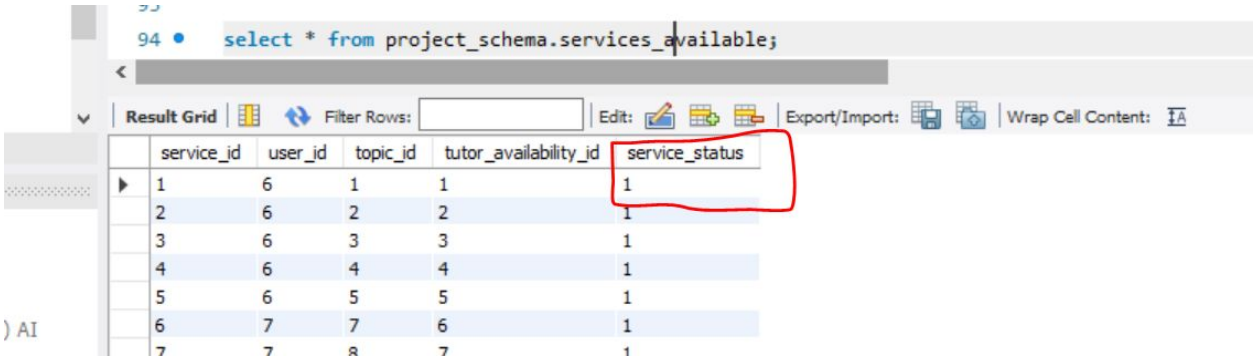
Team Members:

Rahul Ratra **RaviKumar Mittapalli** **Sahithi Priya Gutta** **Noor Zahara**

6. When service is confirmed

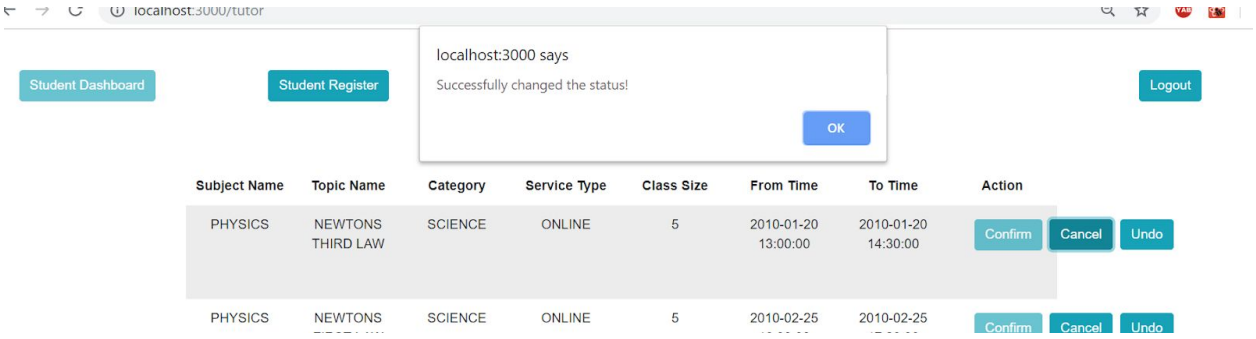
service_status = 1





7. When service is cancelled

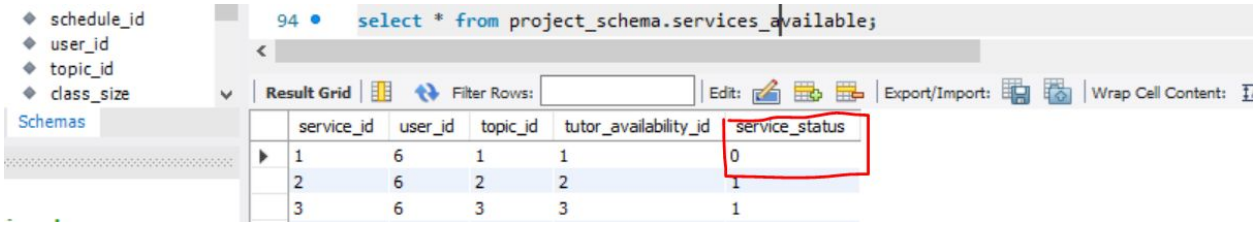
service_status = 0



Project Group 9 - ShareNLearn

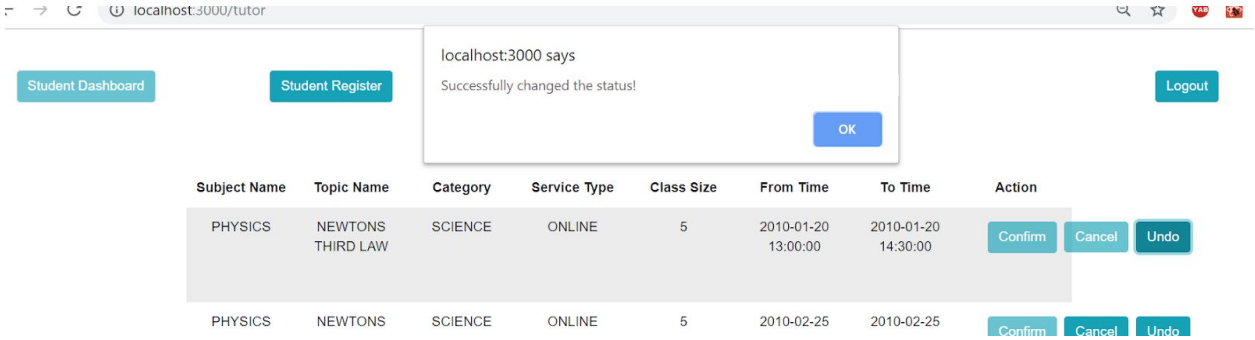
Team Members:

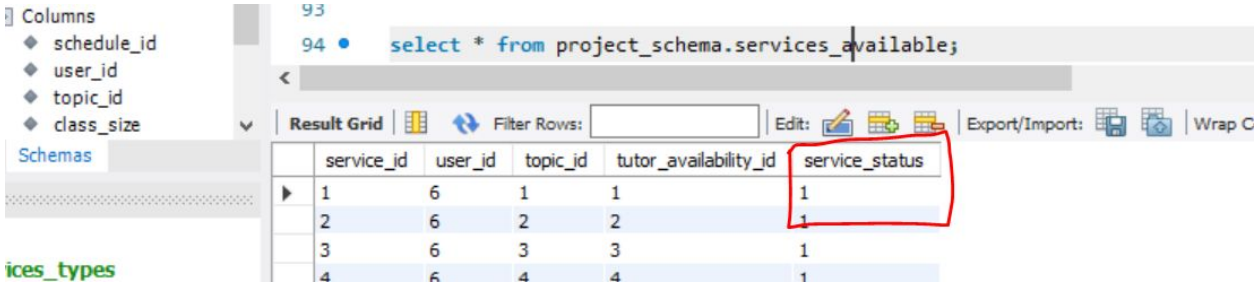
Rahul Ratra **RaviKumar Mittapalli** **Sahithi Priya Gutta** **Noor Zahara**



8. Service status - Undo Functionality

a. When service is cancelled- the status goes to confirmed state during undo operation



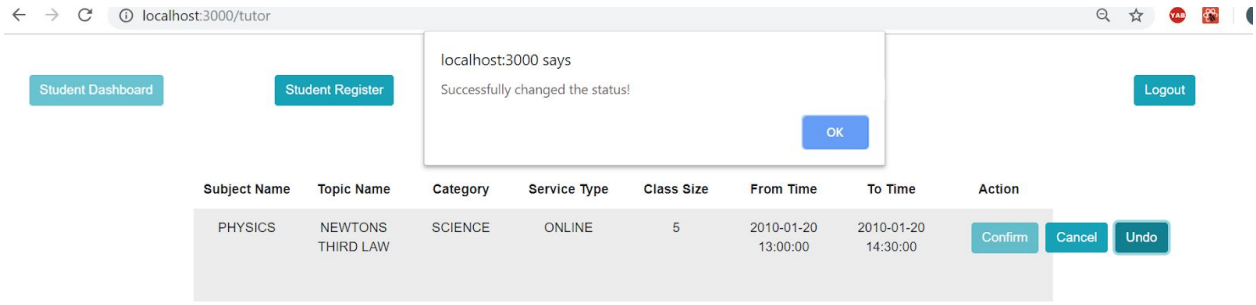


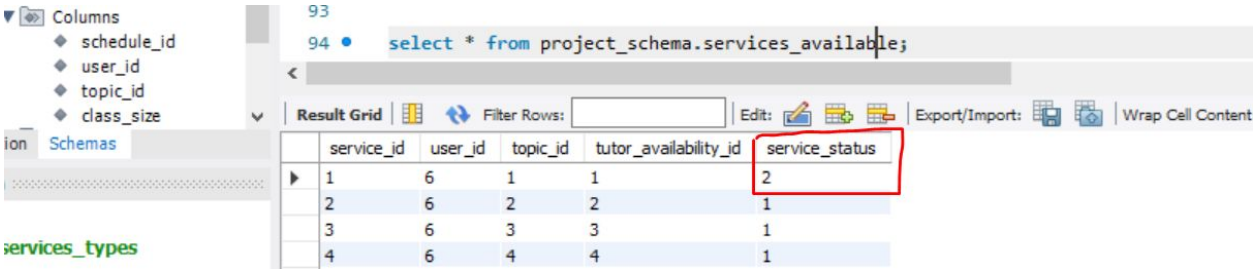
Project Group 9 - ShareNLearn

Team Members:

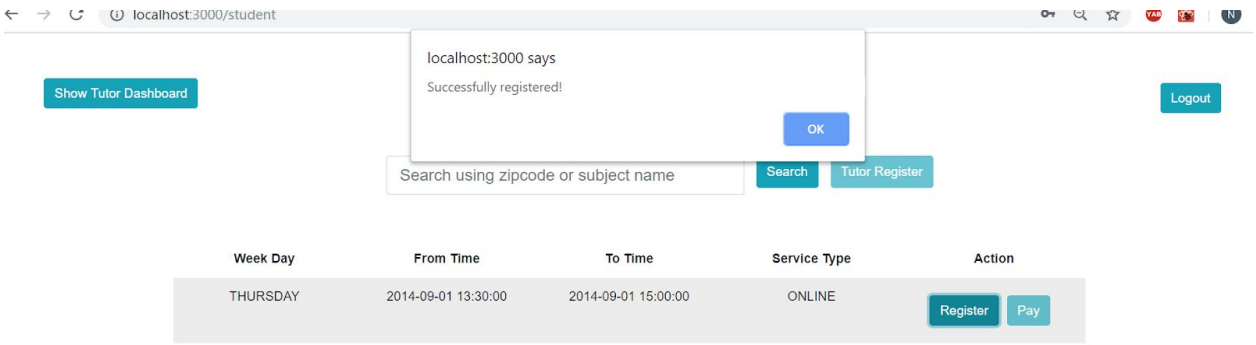
Rahul Ratra **RaviKumar Mittapalli** **Sahithi Priya Gutta** **Noor Zahara**

b. When the service is confirmed , the status changes to not confirmed upon undo operation





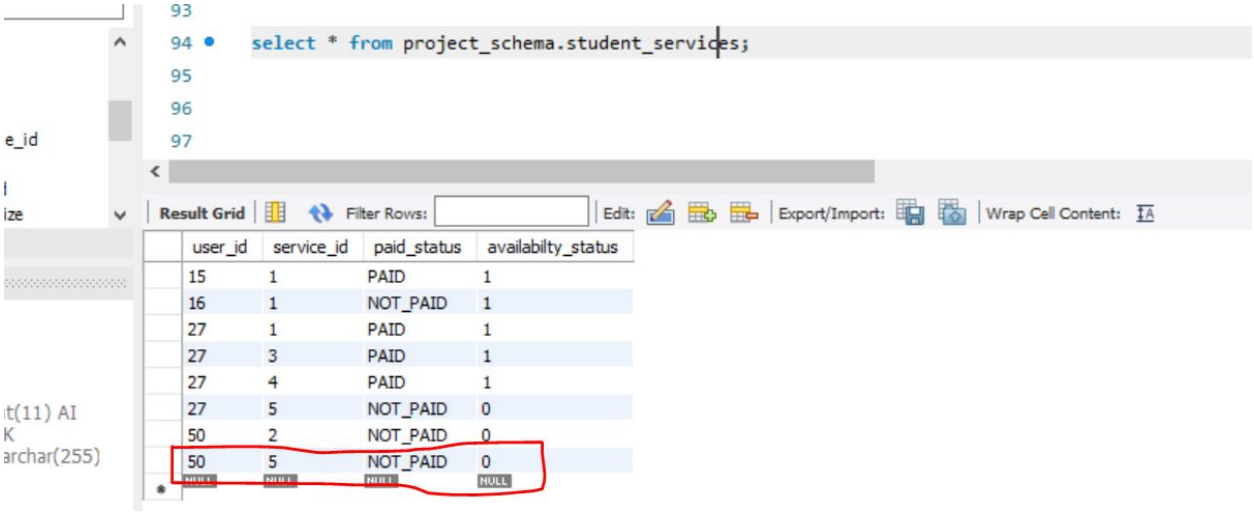
9. Student register for service



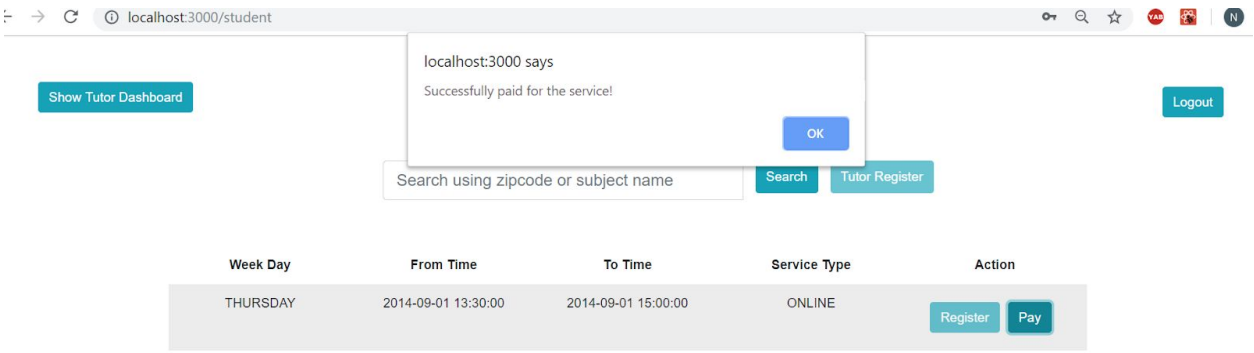
Project Group 9 - ShareNLearn

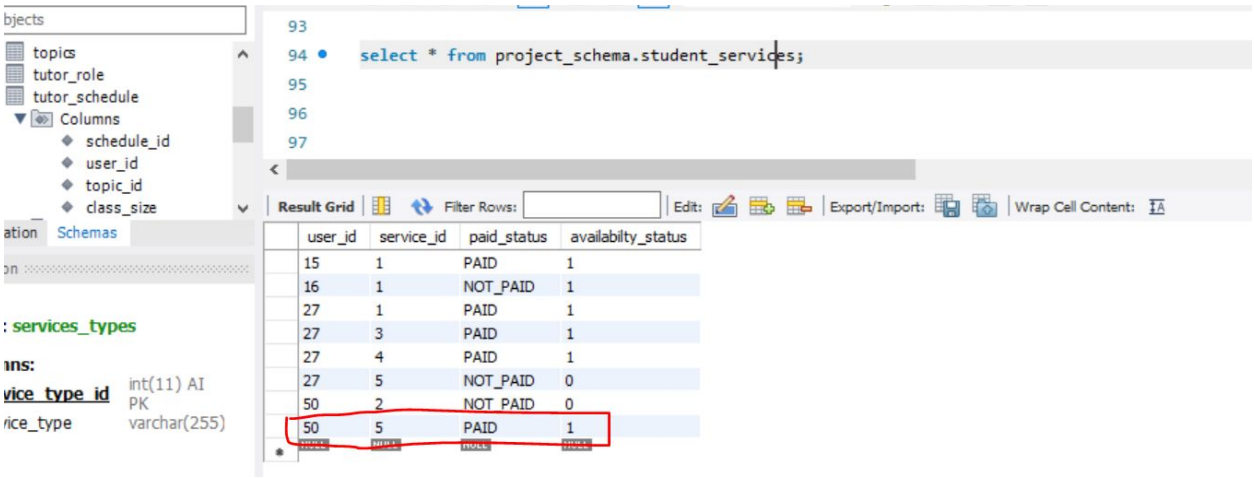
Team Members:

Rahul Ratra **RaviKumar Mittapalli** **Sahithi Priya Gutta** **Noor Zahara**



10. Student pays for the service.





Project Group 9 - ShareNLearn

Team Members:

Rahul Ratra **RaviKumar Mittapalli** **Sahithi Priya Gutta** **Noor Zahara**

Sprint 3

User Stories Considered for this Sprint:

Story ID	Description
US1	User as a tutor can add a service.

Our Conceptual Design and Logical Design are same as in Sprint 2. No changes are done in the current sprint (i.e. Sprint 3).

SQL QUERIES

1. alter table services_available modify column service_status tinyint(1) default 2;
2. insert into tutor_schedule (user_id,topic_id,class_size)values (?,?,?);
3. Insert into availability_timigs(schedule_id,service_type_id,week_day,avail_time_from,avail_time_to) values(?,?,?,?,?);
4. insert into services_available(user_id,topic_id,tutor_availability_id) values (?,?,?);

INDEXES

Note : In INNODB, all the foreign keys are indexes of the corresponding tables automatically.

1. **Table - users**

   Clustered Index: user_id

   User_id is the primary key in the users table and data in the table are ordered in the same way as the primary key. So, creating an index for this column makes search easy.

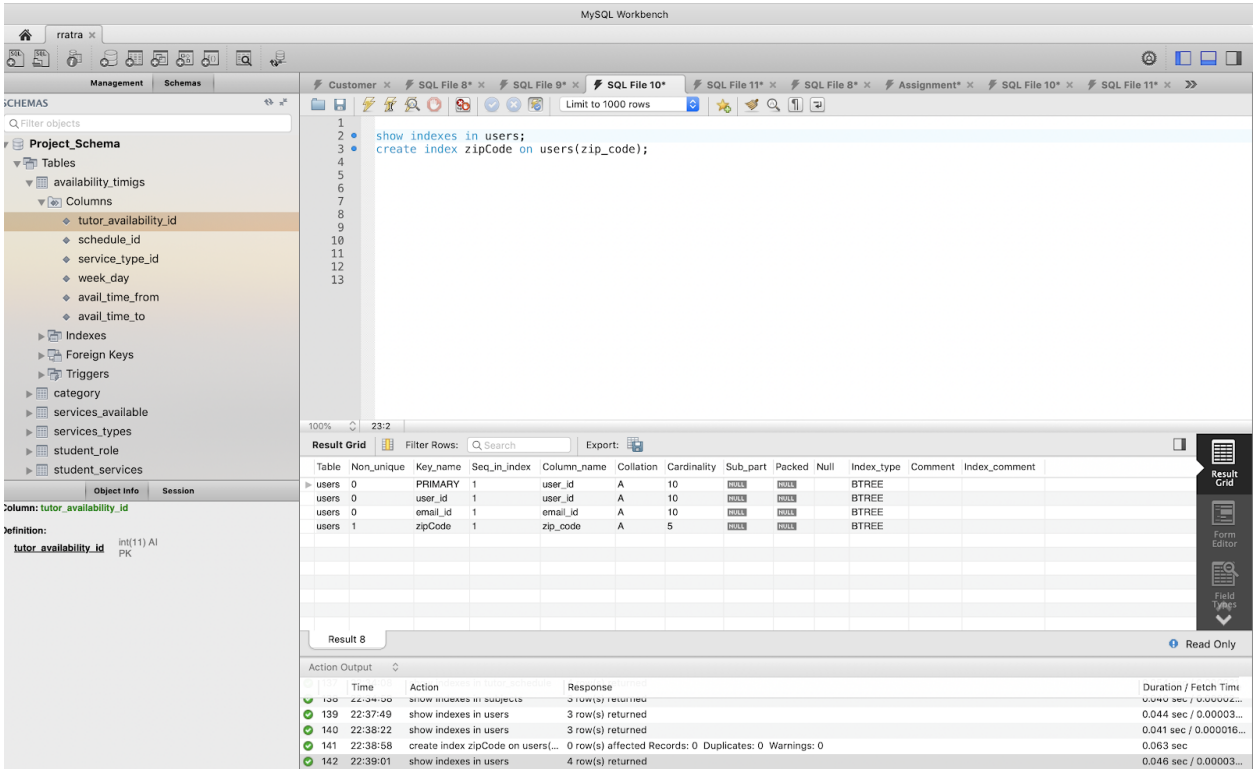
Project Group 9 - ShareNLearn

Team Members:

Rahul Ratra **RaviKumar Mittapalli** **Sahithi Priya Gutta** **Noor Zahara**

Non-Clustered Index:

create index zipCode on users(zip_code);
show index from project_schema.users;



Justification:

Often search for zip_code is made in users table, so creating indexes for this reduces the searching time and enhances the overall performance of the queries.

Project Group 9 - ShareNLearn

Team Members:

Rahul Ratra **RaviKumar Mittapalli** **Sahithi Priya Gutta** **Noor Zahara**

2. Table - subjects

Clustered Index: subject_id

Subject_id are the primary keys in subjects table and data in the table are ordered in the same way as the primary key. So, creating an index for this column makes search easy.

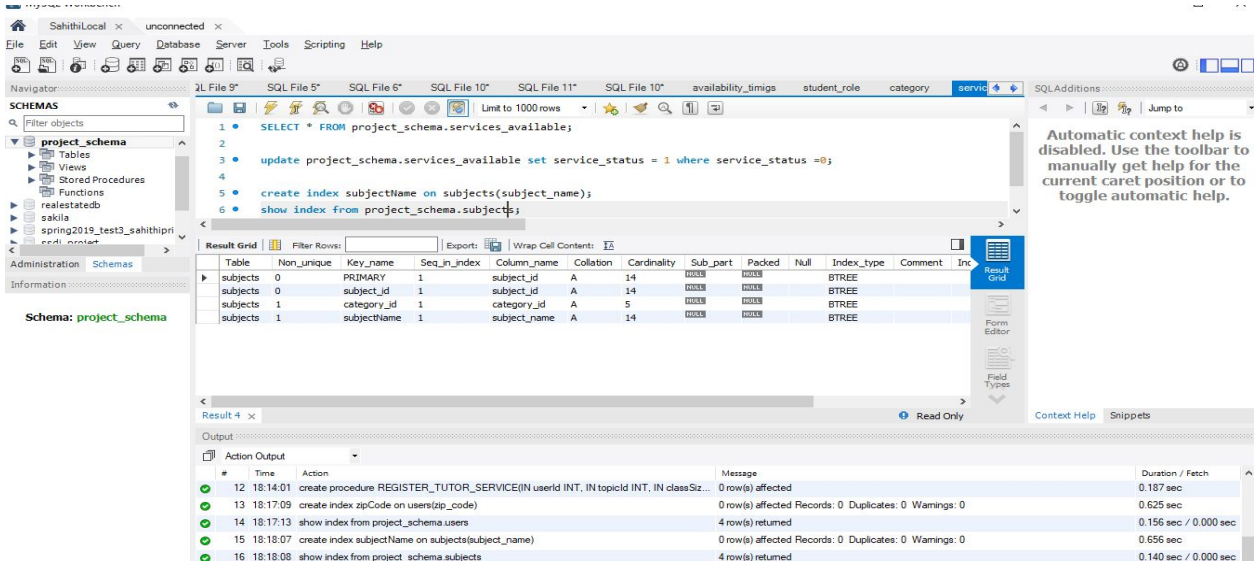
Non-Clustered Index: category_id (Foreign Key)

Explicitly created for subject name as well:

```
create index subjectName on  subjects(subject_name);
show index from project_schema.subjects;
```

Justification:

Often search for subject_name is made in subjects table, so creating indexes for this reduces the searching time and enhances the overall performance of the queries.



Project Group 9 - ShareNLearn

Team Members:

Rahul Ratra **RaviKumar Mittapalli** **Sahithi Priya Gutta** **Noor Zahara**

3. Table - tutor_schedule

Clustered Index: schedule_id

Non- Clustered: user_id,topic_id(Foreign Key)

schedule_id is the primary keys in tutor_schedule table and data in the table are ordered in the same way as the primary key. So, creating an index for this column makes search easy.

Justification:

While showing a particular tutor schedule, queries are being executed based on schedule_id, user_id and topic_id, so creating indexes for this reduces the searching time and enhances the overall performance of the queries.

4. Table - student_services

Clustered Index: user_id, service_id

Note: In our implementation we have selected primary key as a combination of user_id and service_id.

user_id and service_id are the primary keys in student_services table and data in the table are ordered in the same way as the primary key. So, creating an index for this column makes search easy.

Justification:

While fetching tutor schedule details and available service details, queries are being executed which uses user_id and service_id, so creating indexes for this reduces the searching time and enhances the overall performance of the queries.

5. Table - category

Clustered Index: category_id

category_id is the primary key in category table and data in the table are ordered in the same way as the primary key. So, creating an index for this column makes search easy.

Team Members:

Rahul Ratra **RaviKumar Mittapalli** **Sahithi Priya Gutta** **Noor Zahara**

6. Table - topics

Clustered Index: topic_id

topic_id is the primary key in topics table and data in the table are ordered in the same way as the primary key. So, creating an index for this column makes search easy.

7. Table - tutor_role

Clustered Index: user_id

user_id is the primary key in tutor_role table and data in the table are ordered in the same way as the primary key. So, creating an index for this column makes search easy.

8. Table - student_role

Clustered Index: user_id

user_id is the primary key in student_role table and data in the table are ordered in the same way as the primary key. So, creating an index for this column makes search easy.

9. Table - availability_timigs

Clustered Index: tutor_availability_id

Non Clustered:schedule_id,service_type_id (Foreign keys)

Tutor_availability_id is the primary key in availability_timings table and data in the table are ordered in the same way as the primary key. So, creating an index for this column makes search easy.

Justification:

While showing a particular timing (availability timing), queries are being executed based on tutor_availability_id, service_type_id and schedule_id, so creating indexes for this reduces the searching time and enhances the overall performance of the queries.

Project Group 9 - ShareNLearn

Team Members:

Rahul Ratra **RaviKumar Mittapalli** **Sahithi Priya Gutta** **Noor Zahara**

VIEWS

1. **Search for Available Services**

```
create view searchServices as
select usr.first_name, usr.last_name,sb.subject_name,sb.subject_id,
 tp.topic_id, tp.topic_name,sa.tutor_availability_id,usr.zip_code,sa.service_id,sa.service_status
 from services_available sa join users usr on sa.user_id = usr.user_id
inner join tutor_role ur on ur.user_id = usr.user_id
 join tutor_schedule ts on ts.user_id = usr.user_id  join topics tp on tp.topic_id = ts.topic_id
join subjects sb on sb.subject_id = tp.subject_id;
```

Query: select first_name, last_name,subject_name,subject_id, topic_id, topic_name,tutor_availability_id,zip_code,service_id from searchServices where service_status = 1

2. **Fetch Tutor Services**

```
create view getTutorServiceDetails as
select topic_name, subject_name,category_type,avail_time_from, avail_time_to, service_type,
class_size,at.tutor_availability_id,sa.service_status,sa.service_id,ts.user_id from tutor_schedule
ts join topics t on t.topic_id = ts.topic_id
join subjects s on s.subject_id = t.subject_id join category c on c.category_id = s.category_id

join availability_timigs at on at.schedule_id = ts.schedule_id join services_types st on
st.service_type_id = at.service_type_id join services_available sa on sa.user_id = ts.user_id
 and at.tutor_availability_id = sa.tutor_availability_id;
```

Query: select topic_name, subject_name,category_type,avail_time_from, avail_time_to, service_type, class_size,at.tutor_availability_id,sa.service_status,sa.service_id from getTutorServiceDetails where user_id = ?

Project Group 9 - ShareNLearn

Team Members:

Rahul Ratra **RaviKumar Mittapalli** **Sahithi Priya Gutta** **Noor Zahara**

STORED PROCEDURES

1. Procedure Implemented for User Registration

```
DELIMITER $$
CREATE PROCEDURE REGISTER_USER (IN firstname VARCHAR(255), IN lastName VARCHAR(255),
IN email_id VARCHAR(255),
IN user_password VARCHAR(255), IN addressLine1 VARCHAR(255), IN addressLine2 VARCHAR(255),
IN city VARCHAR(255),
IN state VARCHAR(255), IN  country VARCHAR(255), IN zipCode VARCHAR(255), IN userType
VARCHAR(255), OUT result int)
BEGIN
        Declare user_ids int default -1;
        insert into users (first_name,
last_name,email_id,user_password,address_line1,address_line2,city,state,country,zip_code)
  values (firstname, lastName, email_id, user_password, addressLine1, addressLine2, city, state,
country, zipCode);
  select u.user_id  into @user_ids from users u where u.email_id = email_id limit 1;
        if userType = "S" then
                insert into student_role (user_id) values(@user_ids);
  elseif userType = "T" then
                insert into tutor_role (user_id) values(@user_ids);
  elseif userType = "B" then
                insert into student_role (user_id) values(@user_ids);
     insert into tutor_role (user_id) values(@user_ids);
  end if;

  SET result = @user_ids;
END $$
DELIMITER ;
```

2. Procedure Implemented for Tutor Service Registration

```
DELIMITER $$
create procedure REGISTER_TUTOR_SERVICE(IN userId INT, IN topicId INT, IN classSize INT, IN
weekDay VARCHAR(255), IN fromDate VARCHAR(255),
      IN toDate VARCHAR(255), IN serviceType VARCHAR(255))
begin
DECLARE scheduleId int;
DECLARE tutorAvailabilityId int;
insert into tutor_schedule (user_id,topic_id,class_size)VALUES(userId,topicId, classSize);
```

Project Group 9 - ShareNLearn

Team Members:

Rahul Ratra **RaviKumar Mittapalli** **Sahithi Priya Gutta** **Noor Zahara**

```
select schedule_id  into scheduleId from tutor_schedule where user_id = userId and topic_id = topicId and
class_size = classSize order by schedule_id desc limit 1;
insert into
availability_timigs(schedule_id,service_type_id,week_day,avail_time_from,avail_time_to)values(scheduleI
d,serviceType,weekDay,fromDate,toDate);
select tutor_availability_id into tutorAvailabilityId from availability_timigs
where schedule_id = scheduleId
  and service_type_id = serviceType and week_day = weekDay and avail_time_from = fromDate and
avail_time_to = toDate;
insert into services_available(user_id,topic_id,tutor_availability_id)values(userId, topicId,
tutorAvailabilityId);
end $$
DELIMITER ;
```

UI DEMO

User as a Tutor can add a Service: