

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

Project Group 9 - ShareNLearn

Team Members:

Rahul Ratra

RaviKumar Mittapalli

Sahithi Priya Gutta

Noor Zahara

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]

Project Group 9 - ShareNLearn

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

localhost:4200/login

localhost:4200 says true

OK

| | | | | | | | | | | |
|------|-------|----------------|---------------------------|----|-----------|----------------|---------------|-------|---|----------|
| Ravi | Kumar | ravi@gmail.com | 10004 Graduate Lane Apt B | NA | Charlotte | North Carolina | United States | 28262 | S | Register |
|------|-------|----------------|---------------------------|----|-----------|----------------|---------------|-------|---|----------|

student user id zipcode subject Search

Entry in users table

Query 1

```
SELECT * FROM services_available;
SELECT * FROM users;
SELECT * FROM user_role;
```

| user_id | first_name | middle_name | last_name | email_id | address_line1 | address_line2 | city | state | country | zip_code |
|---------|------------|-------------|------------|---------------------|---------------------------|---------------|-----------|----------------|---------------|----------|
| 5 | Sham | | Shetty | sham@gmail.com | 10005 Graduate Lane Apt B | NA | Charlotte | NC | USA | 28264 |
| 6 | Rahul | | Ratra | ratra@uncc.edu | 10004 Graduate Lane Apt B | NA | Charlotte | North Carolina | United States | 28262 |
| 7 | Ravikumar | | Mittapalli | rmmravi@exangle.com | 430 Barton Creek | NA | Charlotte | NA | USA | 28262 |
| 8 | Ravi | | Kumar | ravi@gmail.com | 10004 Graduate Lane Apt B | NA | Charlotte | North Carolina | United States | 28262 |

category 17 tutor_schedule 25 availability_timings 31 users 86

Output

| # | Time | Action | Message | Duration / Fetch |
|-----|----------|----------------------------------|-------------------|-----------------------|
| 120 | 17:21:24 | select * from student_services | 3 row(s) returned | 0.015 sec / 0.000 sec |
| 121 | 17:21:27 | select * from services_available | 2 row(s) returned | 0.000 sec / 0.000 sec |
| 122 | 17:22:04 | select * from student_services | 3 row(s) returned | 0.000 sec / 0.000 sec |
| 123 | 17:22:27 | select * from student_services | 3 row(s) returned | 0.000 sec / 0.000 sec |
| 124 | 17:22:55 | select * from student_services | 3 row(s) returned | 0.000 sec / 0.000 sec |
| 125 | 17:53:02 | select * from services_available | 2 row(s) returned | 0.000 sec / 0.000 sec |
| 126 | 18:15:56 | select * from user_role | 4 row(s) returned | 0.000 sec / 0.000 sec |
| 127 | 18:16:09 | SELECT * from users | 5 row(s) returned | 0.000 sec / 0.000 sec |
| 128 | 18:39:21 | SELECT * from users | 6 row(s) returned | 0.000 sec / 0.000 sec |
| 129 | 18:43:01 | SELECT * from users | 7 row(s) returned | 0.000 sec / 0.000 sec |

Project Group 9 - ShareNLearn

Team Members:

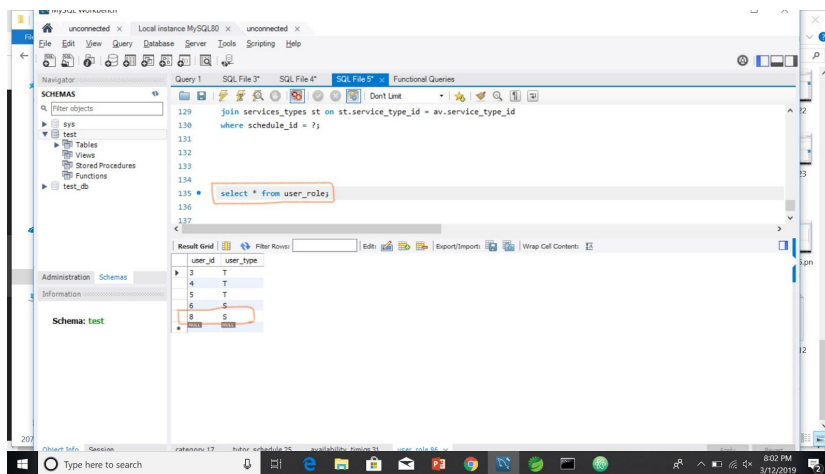
Rahul Ratra

RaviKumar Mittapalli

Sahithi Priya Gutta

Noor Zahara

Entry in User_Role Table



2. User search for available courses

localhost:4200/login

| | | | | | | | | | | |
|------|-------|--------------|---------|----------------|-----|----|---------------------------------------|----------------|---------------|----------|
| Date | Name | Email | DOB | Grade Level | Age | NA | City | State | Country | Zip Code |
| | Kumar | kk@gmail.com | 2004 | Grade Nine Apt | | NA | Charlotte | North Carolina | United States | 28262 |
| S | | 20262 | Subject | | | | <input type="button" value="Search"/> | | | |

| First Name | Last Name | Subject Name | Topic Name | Zip Code |
|------------|-----------|------------------|---------------------|----------|
| Sudip | Acharya | Computer Science | Intelligent Systems | 28262 |
| Anjya | Pandey | Computer Science | Intelligent Systems | 28262 |

Project Group 9 - ShareNLearn

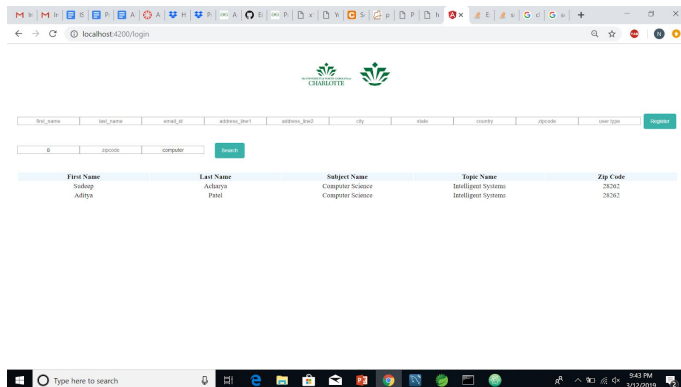
Team Members:

Rahul Ratra

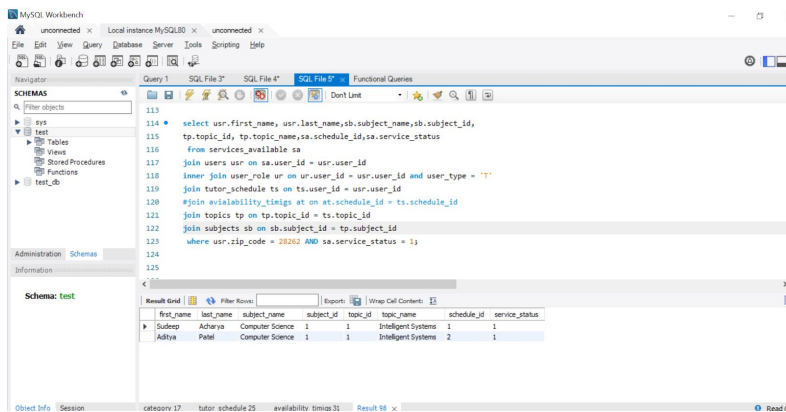
RaviKumar Mittapalli

Sahithi Priya Gutta

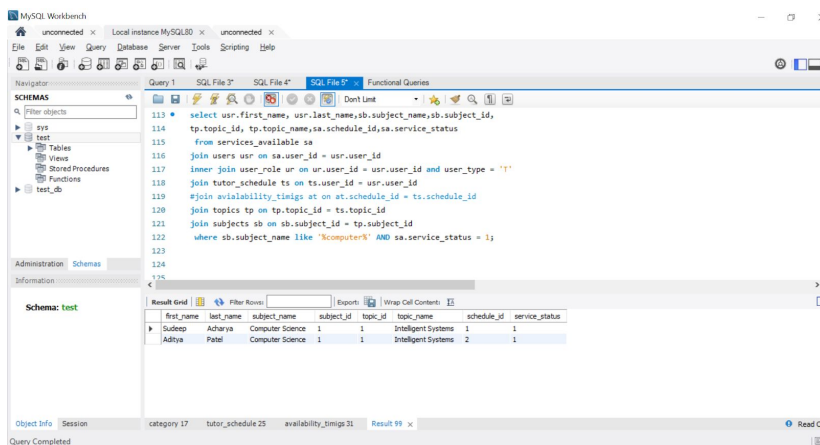
Noor Zahara



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

localhost:4200/login



First Name: Last Name: Email: Password: Confirm Password:

| From Time | To Time | Service Type | Action |
|---------------------|---------------------|--------------|---|
| 2018-03-12 14:30:00 | 2018-03-12 16:30:00 | online | <input type="button" value="Unregister"/> <input type="button" value="Register (Pay)"/> |
| 2018-03-12 17:30:00 | 2018-03-12 19:30:00 | online | <input type="button" value="Unregister"/> <input type="button" value="Register (Pay)"/> |

| First Name | Last Name | Subject Name | Topic Name | Zip Code |
|------------|-----------|------------------|---------------------|----------|
| John | Doe | Computer Science | Intelligent Systems | 75242 |
| Aditya | Patel | Computer Science | Intelligent Systems | 28262 |

Query used for Fetching tutor availability

MySQL Workbench

Query 1

```
119 #join availability_timings at on at.schedule_id = ts.schedule_id
120 join topics tp on tp.topic_id = ts.topic_id
121 join subjects sb on sb.subject_id = tp.subject_id
122 where sb.subject_name like 'Computer%' AND sa.service_status = 1;
123
124
125
126
127 select av.*, st.service_type from availability_timings av
128 join services_types st on st.service_type_id = av.service_type_id
129 where schedule_id = 1;
130
131
```

Result Grid

| tutor_availability_id | schedule_id | service_type_id | week_day | avail_time_from | avail_time_to | service_type |
|-----------------------|-------------|-----------------|----------|---------------------|---------------------|--------------|
| 1 | 1 | 1 | MON | 2018-03-12 14:30:00 | 2018-03-12 16:30:00 | online |
| 2 | 1 | 1 | TUE | 2018-03-12 13:30:00 | 2018-03-12 15:30:00 | online |

3. Register for available services

Project Group 9 - ShareNLearn

Team Members:

Rahul Ratra

RaviKumar Mittapalli

Sahithi Priya Gutta

Noor Zahara

localhost4200/login

localhost4200 says
true

OK

| first_name | last_name | email_id | address_line1 | address_line2 | city | state | country | zipcode | user_type |
|------------|-----------|----------|---------------|---------------|------|-------|---------|---------|-----------|
| 8 | 28262 | subject | | | | | | | |

Search

| From Time | To Time | Service Type | Action |
|---------------------|---------------------|--------------|-------------------|
| 2018-03-12 14:30:00 | 2018-03-12 16:30:00 | online | 2 Register Pay |
| 2018-03-12 13:30:00 | 2018-03-12 15:30:00 | online | 2 Register Pay |

| First Name | Last Name | Subject Name | Topic Name | Zip Code |
|------------|-----------|------------------|---------------------|----------|
| Sudheep | Acharya | Computer Science | Intelligent Systems | 28262 |
| Aditya | Patel | Computer Science | Intelligent Systems | 28262 |

DB Entry post registration

Query 1

```
SELECT * FROM student_services;
SELECT * FROM services_available;
SELECT * FROM users;
SELECT * FROM user_role;
SELECT * FROM services_available;
SELECT * FROM student_services;
```

Result Grid

| user_id | service_id | paid_status | availability_status |
|---------|------------|-------------|---------------------|
| 2 | 1 | PAID | 1 |
| 6 | 1 | PAID | 1 |
| 6 | 2 | NOT_PAID | 1 |
| 8 | 1 | NOT_PAID | 1 |
| 8 | 2 | NOT_PAID | 1 |

4. User pays for the services availed

localhost4200/login

localhost4200 says
true

OK

| first_name | last_name | email_id | address_line1 | address_line2 | city | state | country | zipcode | user_type |
|------------|-----------|----------|---------------|---------------|------|-------|---------|---------|-----------|
| 8 | 28262 | subject | | | | | | | |

Search

| From Time | To Time | Service Type | Action |
|---------------------|---------------------|--------------|-------------------|
| 2018-03-12 14:30:00 | 2018-03-12 16:30:00 | online | 2 Register Pay |
| 2018-03-12 13:30:00 | 2018-03-12 15:30:00 | online | 2 Register Pay |

| First Name | Last Name | Subject Name | Topic Name | Zip Code |
|------------|-----------|------------------|---------------------|----------|
| Sudheep | Acharya | Computer Science | Intelligent Systems | 28262 |
| Aditya | Patel | Computer Science | Intelligent Systems | 28262 |

DB Update post payment

Project Group 9 - ShareNLearn

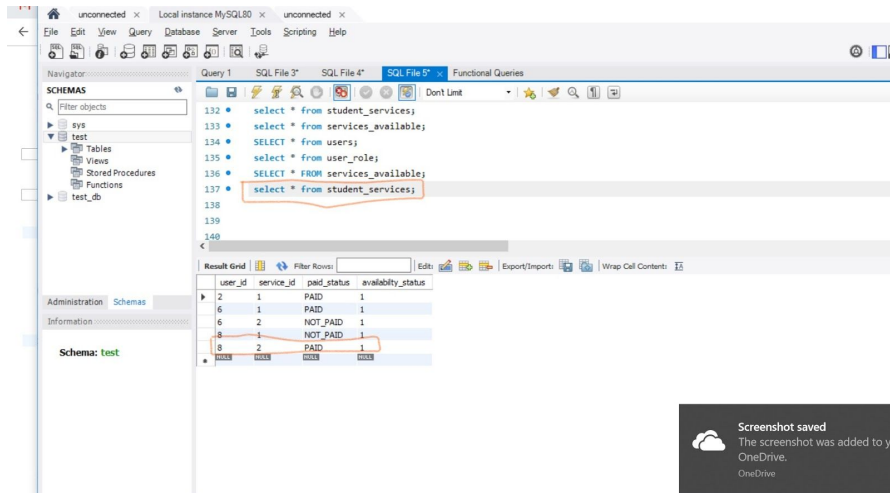
Team Members:

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.

Project Group 9 - ShareNLearn

Team Members:

Rahul Ratra RaviKumar Mittapalli Sahithi Priya Gutta Noor Zahara

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

Project Group 9 - ShareNLearn

Team Members:

Rahul Ratra RaviKumar Mittapalli Sahithi Priya Gutta Noor Zahara

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_timings 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_timings 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

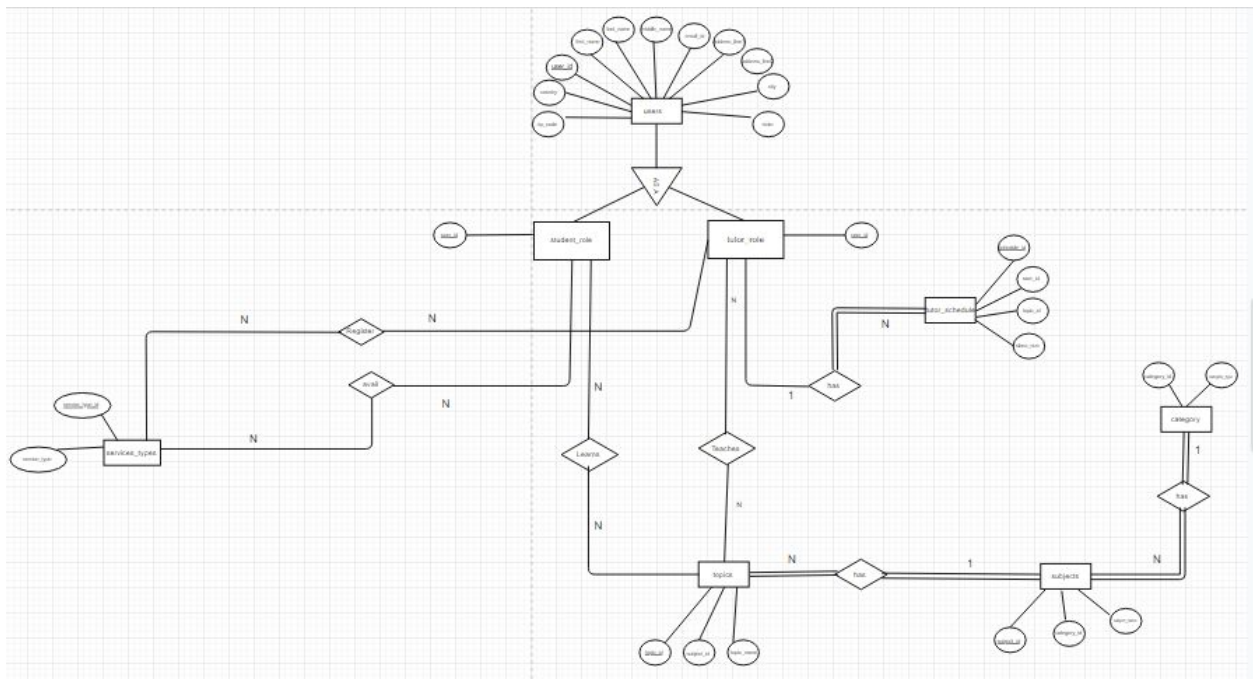
Sahithi Priya Gutta

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

→ ↻ ⓘ localhost:3000 🔑 🔍 ☆

localhost:3000 says
User Id created :49

OK

Register Login

Sahithi

Gutta

sahithi@gmail.com

10004 Graduate Lane Apt B

NA

Charlotte

North Carolina

United States

28262

S

Send

← → ↻ ⓘ localhost:3000 🔑 🔍 ☆ 📄 📄 📄

localhost:3000 says
User Id created :50

OK

Register Login

Madhuri

N

madhuri@gmail.com

Graduate Lane, 9523

NA

Charlotte

North Carolina

United States

28262

T

Send

Project Group 9 - ShareNLearn

Team Members:

Rahul Ratra

RaviKumar Mittapalli

Sahithi Priya Gutta

Noor Zahara

localhost:3000

localhost:3000 says
User Id created :51

OK

Register Login

Sowmya

Sundar

sowmya@gmail.com

Graduate Lane 10005

NA

Charlotte

North Carolina

United States

28262

B

Send

97

98 • select * from project_schema.users;

99

Result Grid

| user_id | first_name | middle_name | last_name | email_id | password | address_line1 | address_line2 | city | state | country | zip_code |
|---------|------------|-------------|-----------|--------------------|----------|---------------------------|---------------|-----------|----------------|------------|----------|
| 47 | Noor | | Zahara | noor19zahara@gm... | qwerty | 10004 Graduate Lane Apt B | | Charlotte | North Carolina | United ... | 28262 |
| 48 | Noor | | Zahara | noor16zahara@gm... | qwerty | 10004 Graduate Lane Apt B | | Charlotte | North Carolina | United ... | 28262 |
| 49 | Sahithi | | Gutta | sahithi@gmail.com | 1234 | 10004 Graduate Lane Apt B | NA | Charlotte | North Carolina | United ... | 28262 |
| 50 | Madhuri | | N | madhuri@gmail.com | 1234 | Graduate Lane, 9523 | NA | Charlotte | North Carolina | United ... | 28262 |
| 51 | Sowmya | | Sundar | sowmya@gmail.com | welcome | Graduate Lane 10005 | NA | Charlotte | North Carolina | United ... | 28262 |
| * | | | | | | | | | | | |

users 19 x

Output

99 • select * from project_schema.student_roles;

100

Result Grid

| user_id |
|---------|
| 46 |
| 47 |
| 48 |
| 49 |
| 51 |
| * |

Project Group 9 - ShareNLearn

Team Members:

Rahul Ratra

RaviKumar Mittapalli

Sahithi Priya Gutta

Noor Zahara

The screenshot shows a database management interface. On the left, a schema diagram for 'project_schema' is visible, showing a table 'tutor_role' with columns 'user_id' (int(11) AI PK) and 'vice_type' (varchar(255)). The main area displays a SQL query: `select * from project_schema.tutor_role;`. Below the query, a 'Result Grid' shows the following data:

| user_id |
|---------|
| 45 |
| 46 |
| 47 |
| 48 |
| 50 |
| NULL |

The interface also includes a 'Filter Rows' section and a 'Wrap Cell Content' option.

2. Login Feature

- As a student

Project Group 9 - ShareNLearn

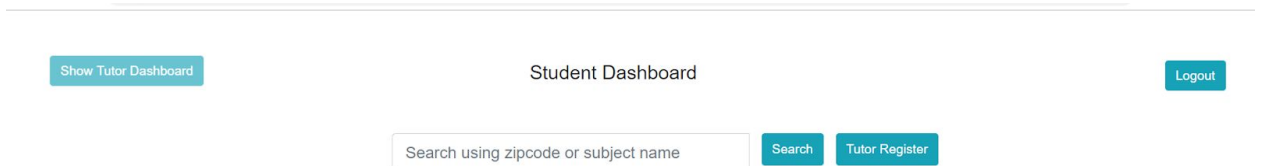
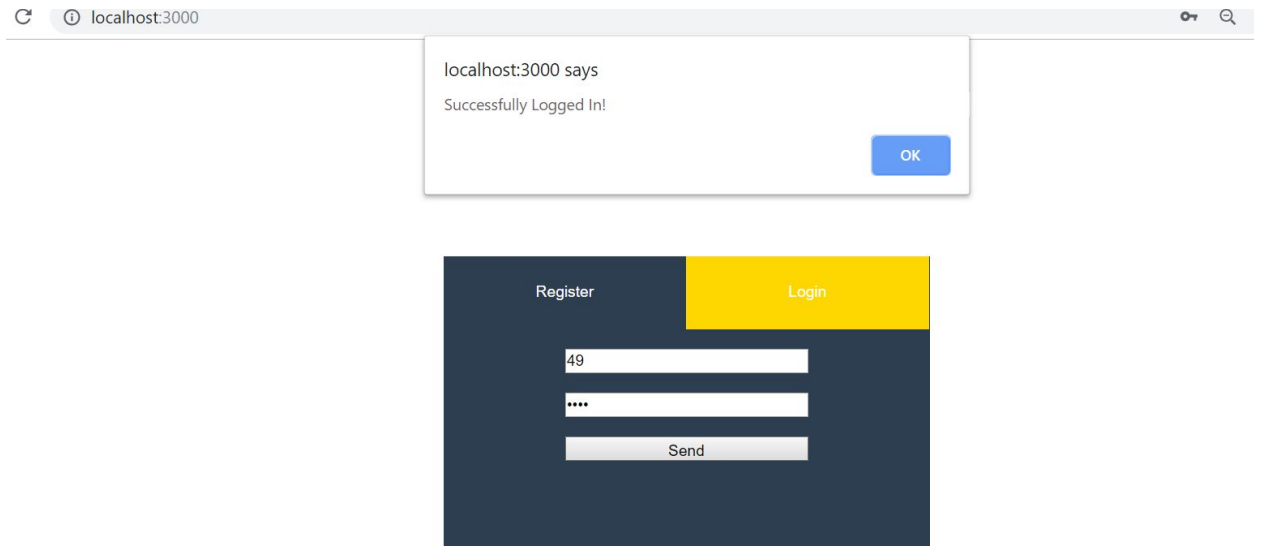
Team Members:

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

[Student Dashboard](#)[Student Register](#)[Logout](#)

Tutor Dashboard

| Subject Name | Topic Name | Category | Service Type | Class Size | From Time | To Time | Action |
|--------------|-------------------|----------|--------------|------------|---------------------|---------------------|---|
| PHYSICS | NEWTONS THIRD LAW | SCIENCE | ONLINE | 5 | 2010-01-20 13:00:00 | 2010-01-20 14:30:00 | Confirm Cancel Undo |
| PHYSICS | NEWTONS FIRST LAW | SCIENCE | ONLINE | 5 | 2010-02-25 16:30:00 | 2010-02-25 17:20:00 | Confirm Cancel Undo |
| PHYSICS | NEWTONS SECOND | SCIENCE | ONLINE | 5 | 2011-11-13 14:00:00 | 2011-11-13 15:30:00 | Confirm Cancel Undo |

c. As Both

localhost:3000/student

[Show Tutor Dashboard](#)[Logout](#)

Student Dashboard

[Search](#)[Tutor Register](#)

3. User as a student registers as a Tutor

Project Group 9 - ShareNLearn

Team Members:

Rahul Ratra

RaviKumar Mittapalli

Sahithi Priya Gutta

Noor Zahara

localhost:3000/student

localhost:3000 says
Successfully registered as a Tutor!

Show Tutor Dashboard

Search using zipcode or subject name

Search Tutor Register

Logout

select * from project_schema.tutor_role;

Result Grid

| user_id | class_size |
|---------|------------|
| 47 | 100 |

4. User as a Tutor registers as a Student

localhost:3000/tutor

Student Dashboard

Student Register

localhost:3000 says
Successfully registered as a Student!

Logout

| Subject Name | Topic Name | Category | Service Type | Class Size | From Time | To Time | Action |
|--------------|-------------------|----------|--------------|------------|---------------------|---------------------|---------------------|
| PHYSICS | NEWTONS THIRD LAW | SCIENCE | ONLINE | 5 | 2010-01-20 13:00:00 | 2010-01-20 14:30:00 | Confirm Cancel Undo |
| PHYSICS | NEWTONS FIRST LAW | SCIENCE | ONLINE | 5 | 2010-02-25 16:30:00 | 2010-02-25 17:20:00 | Confirm Cancel Undo |

Project Group 9 - ShareNLearn

Team Members:

Rahul Ratra

RaviKumar Mittapalli

Sahithi Priya Gutta

Noor Zahara

The screenshot shows a database management interface. On the left, there's a sidebar with a tree view containing 'schemas' and 'services_types'. The main area displays a SQL query: `select * from project_schema.student_role;`. Below the query, a 'Result Grid' shows a table with columns 'user_id' and 'topic_id'. The data rows are: (47, 48), (49, 50), (51, 51), and a row with 'NULL' in the 'topic_id' column. The interface also includes a 'Filter Rows' field and various action buttons like 'Edit', 'Export/Import', and 'Wrap Cell Content'.

5. When service is in not confirmed state

service_status = 2

The screenshot shows a web application interface. At the top, there are buttons for 'Student Dashboard', 'Student Register', and 'Logout'. The main section is titled 'Tutor Dashboard' and contains a table with the following columns: 'Subject Name', 'Topic Name', 'Category', 'Service Type', 'Class Size', 'From Time', 'To Time', and 'Action'. The table has two rows of data. The first row has 'PHYSICS' as the subject, 'NEWTONS THIRD LAW' as the topic, 'SCIENCE' as the category, 'ONLINE' as the service type, '5' as the class size, '2010-01-20 13:00:00' as the from time, and '2010-01-20 14:30:00' as the to time. The 'Action' column for this row contains three buttons: 'Confirm', 'Cancel', and 'Undo'. The second row has 'PHYSICS' as the subject, 'NEWTONS FIRST LAW' as the topic, 'SCIENCE' as the category, 'ONLINE' as the service type, '5' as the class size, '2010-02-25 16:30:00' as the from time, and '2010-02-25 17:20:00' as the to time. The 'Action' column for this row also contains 'Confirm', 'Cancel', and 'Undo' buttons. A red box highlights the 'Confirm', 'Cancel', and 'Undo' buttons in the first row.

The screenshot shows a database management interface. On the left, there's a sidebar with a tree view containing 'schemas' and 'services_types'. The main area displays a SQL query: `select * from project_schema.services_available;`. Below the query, a 'Result Grid' shows a table with columns 'service_id', 'user_id', 'topic_id', 'tutor_availability_id', and 'service_status'. The data rows are: (1, 6, 1, 1, 2), (2, 6, 2, 2, 1), (3, 6, 3, 3, 1), (4, 6, 4, 4, 1), (5, 6, 5, 5, 1), and (6, 7, 7, 6, 1). The 'service_status' column for the first row (1, 6, 1, 1, 2) is highlighted with a red box.

Project Group 9 - ShareNLearn

Team Members:

Rahul Ratra

RaviKumar Mittapalli

Sahithi Priya Gutta

Noor Zahara

6. When service is confirmed

`service_status = 1`

The screenshot shows a web application interface with a navigation bar containing 'Student Dashboard', 'Student Register', and 'Logout' buttons. A modal dialog box displays the message: 'localhost:3000 says Successfully changed the status!' with an 'OK' button. Below the dialog is a table with the following data:

| Subject Name | Topic Name | Category | Service Type | Class Size | From Time | To Time | Action |
|--------------|-------------------|----------|--------------|------------|---------------------|---------------------|---------------------|
| PHYSICS | NEWTONS THIRD LAW | SCIENCE | ONLINE | 5 | 2010-01-20 13:00:00 | 2010-01-20 14:30:00 | Confirm Cancel Undo |

The screenshot shows a SQL query result grid for the query: `select * from project_schema.services_available;`. The result grid has the following columns: `service_id`, `user_id`, `topic_id`, `tutor_availability_id`, and `service_status`. The `service_status` column is highlighted with a red box. The data rows are as follows:

| service_id | user_id | topic_id | tutor_availability_id | service_status |
|------------|---------|----------|-----------------------|----------------|
| 1 | 6 | 1 | 1 | 1 |
| 2 | 6 | 2 | 2 | 1 |
| 3 | 6 | 3 | 3 | 1 |
| 4 | 6 | 4 | 4 | 1 |
| 5 | 6 | 5 | 5 | 1 |
| 6 | 7 | 7 | 6 | 1 |
| 7 | 7 | 8 | 7 | 1 |

7. When service is cancelled

`service_status = 0`

The screenshot shows the same web application interface as before. The modal dialog box displays the message: 'localhost:3000 says Successfully changed the status!' with an 'OK' button. Below the dialog is a table with the following data:

| Subject Name | Topic Name | Category | Service Type | Class Size | From Time | To Time | Action |
|--------------|-------------------|----------|--------------|------------|---------------------|---------------------|---------------------|
| PHYSICS | NEWTONS THIRD LAW | SCIENCE | ONLINE | 5 | 2010-01-20 13:00:00 | 2010-01-20 14:30:00 | Confirm Cancel Undo |
| PHYSICS | NEWTONS | SCIENCE | ONLINE | 5 | 2010-02-25 | 2010-02-25 | Confirm Cancel Undo |

Project Group 9 - ShareNLearn

Team Members:

Rahul Ratra

RaviKumar Mittapalli

Sahithi Priya Gutta

Noor Zahara

The screenshot shows a database management tool interface. On the left, there's a 'Schemas' panel with a tree view containing 'schedule_id', 'user_id', 'topic_id', and 'class_size'. The main area displays a SQL query: `select * from project_schema.services_available;`. Below the query, there's a 'Result Grid' showing the following data:

| service_id | user_id | topic_id | tutor_availability_id | service_status |
|------------|---------|----------|-----------------------|----------------|
| 1 | 6 | 1 | 1 | 0 |
| 2 | 6 | 2 | 2 | 1 |
| 3 | 6 | 3 | 3 | 1 |

The 'service_status' column is highlighted with a red box, showing values 0, 1, and 1 for the three rows.

8. Service status - Undo Functionality

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

The screenshot shows a web application interface. At the top, there's a navigation bar with buttons for 'Student Dashboard', 'Student Register', and 'Logout'. A confirmation dialog box is open, displaying the message: 'localhost:3000 says Successfully changed the status!'. Below the dialog, there's a table with the following columns: 'Subject Name', 'Topic Name', 'Category', 'Service Type', 'Class Size', 'From Time', 'To Time', and 'Action'.

| Subject Name | Topic Name | Category | Service Type | Class Size | From Time | To Time | Action |
|--------------|-------------------|----------|--------------|------------|---------------------|---------------------|---------------------|
| PHYSICS | NEWTONS THIRD LAW | SCIENCE | ONLINE | 5 | 2010-01-20 13:00:00 | 2010-01-20 14:30:00 | Confirm Cancel Undo |
| PHYSICS | NEWTONS | SCIENCE | ONLINE | 5 | 2010-02-25 | 2010-02-25 | Confirm Cancel Undo |

Below the table, there's a 'Columns' panel with a tree view containing 'schedule_id', 'user_id', 'topic_id', and 'class_size'. The main area displays a SQL query: `select * from project_schema.services_available;`. Below the query, there's a 'Result Grid' showing the following data:

| service_id | user_id | topic_id | tutor_availability_id | service_status |
|------------|---------|----------|-----------------------|----------------|
| 1 | 6 | 1 | 1 | 1 |
| 2 | 6 | 2 | 2 | 1 |
| 3 | 6 | 3 | 3 | 1 |
| 4 | 6 | 4 | 4 | 1 |

The 'service_status' column is highlighted with a red box, showing values 1, 1, 1, and 1 for the four rows.

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

The screenshot shows a web application interface. At the top, there's a navigation bar with buttons: "Student Dashboard", "Student Register", and "Logout". A modal dialog box is open, displaying "localhost:3000 says" and "Successfully changed the status!". Below the dialog, there's a table with columns: "Subject Name", "Topic Name", "Category", "Service Type", "Class Size", "From Time", "To Time", and "Action". The table contains one row for "PHYSICS" and "NEWTONS THIRD LAW". The "Action" column has buttons: "Confirm", "Cancel", and "Undo".

Below the table, there's a section for "Columns" and "Schemas". The "Columns" section lists: "schedule_id", "user_id", "topic_id", and "class_size". The "Schemas" section shows a table with columns: "service_id", "user_id", "topic_id", "tutor_availability_id", and "service_status". The "service_status" column is highlighted with a red box. The table contains four rows of data.

Below the table, there's a section for "services_types" with a table containing one row of data.

9. Student register for service

The screenshot shows a web application interface. At the top, there's a navigation bar with buttons: "Show Tutor Dashboard", "Search", "Tutor Register", and "Logout". A modal dialog box is open, displaying "localhost:3000 says" and "Successfully registered!". Below the dialog, there's a search bar with the text "Search using zipcode or subject name".

Below the search bar, there's a table with columns: "Week Day", "From Time", "To Time", "Service Type", and "Action". The table contains one row for "THURSDAY". The "Action" column has buttons: "Register" and "Pay".

Project Group 9 - ShareNLearn

Team Members:

Rahul Ratra

RaviKumar Mittapalli

Sahithi Priya Gutta

Noor Zahara

The screenshot shows a database query interface. The SQL query is: `select * from project_schema.student_services;`. The results are displayed in a table with the following columns: `user_id`, `service_id`, `paid_status`, and `availability_status`. The results are as follows:

| user_id | service_id | paid_status | availability_status |
|---------|------------|-------------|---------------------|
| 15 | 1 | PAID | 1 |
| 16 | 1 | NOT_PAID | 1 |
| 27 | 1 | PAID | 1 |
| 27 | 3 | PAID | 1 |
| 27 | 4 | PAID | 1 |
| 27 | 5 | NOT_PAID | 0 |
| 50 | 2 | NOT_PAID | 0 |
| 50 | 5 | NOT_PAID | 0 |

The last row (50, 5, NOT_PAID, 0) is highlighted with a red box.

10. Student pays for the service.

The screenshot shows a web application interface. A message box displays: "localhost:3000 says Successfully paid for the service!". Below the message box, there is a search bar with the text "Search using zipcode or subject name" and buttons for "Search" and "Tutor Register". At the bottom, there is a table with the following columns: "Week Day", "From Time", "To Time", "Service Type", and "Action". The table contains one row with the following data:

| Week Day | From Time | To Time | Service Type | Action |
|----------|---------------------|---------------------|--------------|--|
| THURSDAY | 2014-09-01 13:30:00 | 2014-09-01 15:00:00 | ONLINE | <button>Register</button> <button>Pay</button> |

The screenshot shows a database query interface. The SQL query is: `select * from project_schema.student_services;`. The results are displayed in a table with the following columns: `user_id`, `service_id`, `paid_status`, and `availability_status`. The results are as follows:

| user_id | service_id | paid_status | availability_status |
|---------|------------|-------------|---------------------|
| 15 | 1 | PAID | 1 |
| 16 | 1 | NOT_PAID | 1 |
| 27 | 1 | PAID | 1 |
| 27 | 3 | PAID | 1 |
| 27 | 4 | PAID | 1 |
| 27 | 5 | NOT_PAID | 0 |
| 50 | 2 | NOT_PAID | 0 |
| 50 | 5 | PAID | 1 |

The last row (50, 5, PAID, 1) is highlighted with a red box.

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_timings(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;

The screenshot displays the MySQL Workbench interface. The left sidebar shows the 'Project_Schema' with a tree view of tables and columns. The 'users' table is selected, showing columns like 'user_id', 'email_id', and 'zip_code'. The main editor window contains the following SQL queries:

```
1 show indexes in users;
2
3 create index zipCode on users(zip_code);
4
5
6
7
8
9
10
11
12
13
```

The 'Result Grid' at the bottom shows the output of the queries. The first query, 'show indexes in users', returns 3 rows. The second query, 'create index zipCode on users(zip_code)', returns 0 rows affected. The third query, 'show indexes in users', returns 4 rows.

| Table | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment | Index_comment |
|-------|------------|----------|--------------|-------------|-----------|-------------|----------|--------|------|------------|---------|---------------|
| users | 0 | PRIMARY | 1 | user_id | A | 10 | | | | BTREE | | |
| users | 0 | | 1 | user_id | A | 10 | | | | BTREE | | |
| users | 0 | email_id | 1 | email_id | A | 10 | | | | BTREE | | |
| users | 1 | zipCode | 1 | zip_code | A | 5 | | | | BTREE | | |

The 'Action Output' pane at the bottom shows the execution details of the queries:

| Time | Action | Response | Duration / Fetch Time |
|----------|------------------------------------|--|-------------------------|
| 22:37:49 | show indexes in users | 3 row(s) returned | 0.044 sec / 0.00003... |
| 22:38:22 | show indexes in users | 3 row(s) returned | 0.041 sec / 0.000016... |
| 22:38:58 | create index zipCode on users(...) | 0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0 | 0.063 sec |
| 22:39:01 | show indexes in users | 4 row(s) returned | 0.046 sec / 0.00003... |

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.

The screenshot displays the SQL Server Enterprise Manager interface. The left pane shows the 'project_schema' database with a tree view of tables, views, stored procedures, functions, and schemas. The right pane shows the 'subjects' table with its columns: subject_id, category_id, and subject_name. The 'Index' tab is selected, showing the following indexes:

| Table | Non-unique | Key_name | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment |
|----------|------------|-------------|--------------|--------------|-----------|-------------|----------|--------|------|------------|---------|
| subjects | 0 | PRIMARY | 1 | subject_id | A | 14 | | | | BTREE | |
| subjects | 0 | subject_id | 1 | subject_id | A | 14 | | | | BTREE | |
| subjects | 1 | category_id | 1 | category_id | A | 5 | | | | BTREE | |
| subjects | 1 | subjectName | 1 | subject_name | A | 14 | | | | BTREE | |

The bottom pane shows the 'Action Output' window with the following results:

| # | Time | Action | Message | Duration / Fetch |
|----|----------|---|--|-----------------------|
| 12 | 18:14:01 | create procedure REGISTER_TUTOR_SERVICE(IN userid INT, IN topicid INT, IN classSiz... | 0 row(s) affected | 0.187 sec |
| 13 | 18:17:09 | create index zipCode on users(zip_code) | 0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0 | 0.625 sec |
| 14 | 18:17:13 | show index from project_schema.users | 4 row(s) returned | 0.156 sec / 0.000 sec |
| 15 | 18:18:07 | create index subjectName on subjects(subject_name) | 0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0 | 0.656 sec |
| 16 | 18:18:08 | show index from project_schema.subjects | 4 row(s) returned | 0.140 sec / 0.000 sec |

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.

Project Group 9 - ShareNLearn

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_timings

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_timings 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(scheduleId
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:

The screenshot shows a web browser at localhost:3000/tutor. The page has a navigation bar with buttons: Student Dashboard, Student Register, Service Register, and Logout. The main content area is titled "Tutor Dashboard". Below the title is a form with the following fields:

| Category | Subject Topic | Service Type | Class Size | From Time | To Time | Action |
|----------|---------------|--------------|------------|-------------------|-------------------|----------|
| SCIENCE | | Online | | 4/25/2019 1:30 PM | 4/25/2019 1:30 PM | Register |