DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

SCHOOL OF TECHNOLOGY

PANDIT DEENDAYAL ENERGY UNIVERSITY

SESSION 2023-24

SUBMITTED BY

NAME	**:**	**Darshan Patel, Harsh Hirani**
ROLL NO.	**:**	**22BCP006, 22BCP036**
DIVISION	**:**	**1 GROUP: 1**
COURSE NAME	**:**	**DBMS LAB**
COURSE CODE	**:**	**20CP208P**

DBMS PROJECT

SUBMITTED TO

Dr. Amitava Choudhury

Assistant Professor

Department of Computer Science and Engineering

Pandit Deendayal Energy University

Project Title: Foody Paradise

Problem Statement:

To Design a database management system of recipe recommendation. The system should allow users to explore various recipe with course and cuisine, view details about each instruction and ingredients needed to make delicious food. Additionally, the system should provide comment system to user with addition feature to edit and delete.

Project Objectives:

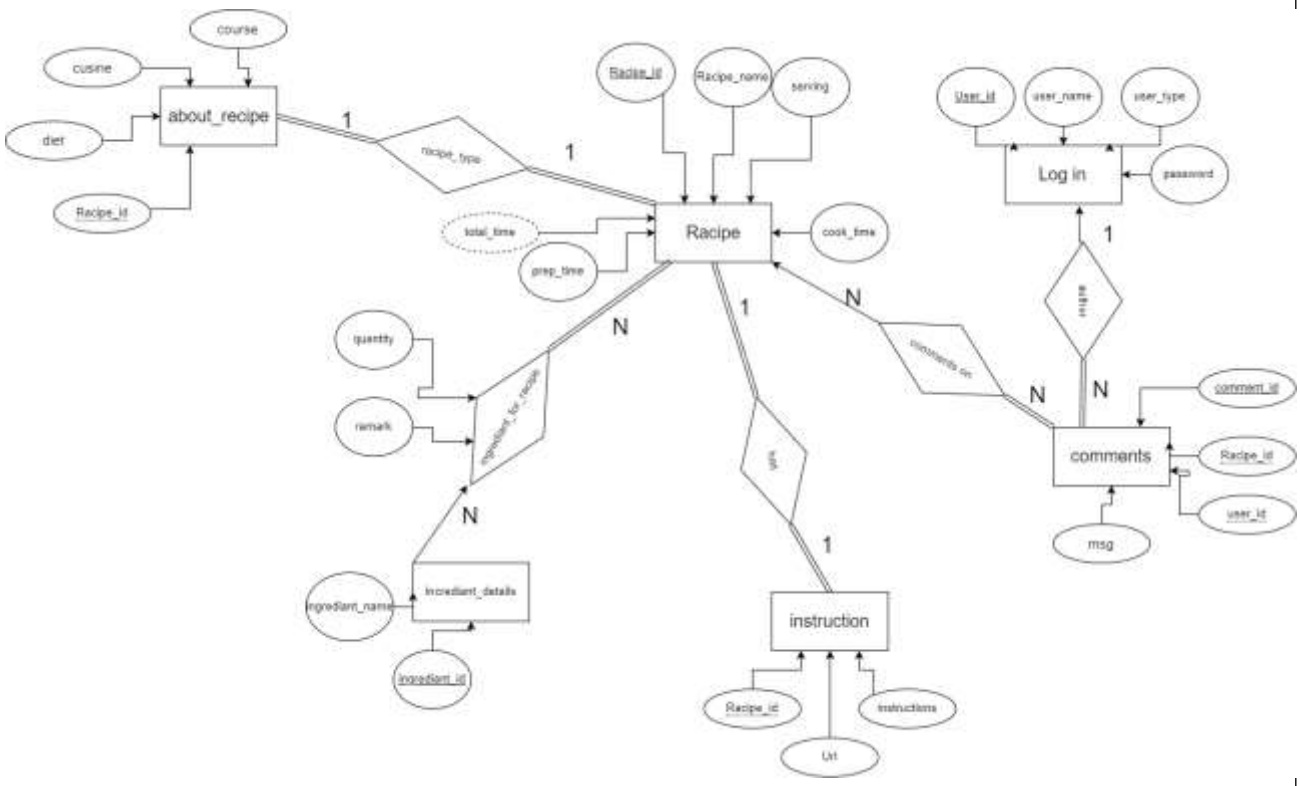
1. User Management:
   - Allow users to register and login to the system securely.
   - Enable users to view their comment on recipe and change or delete.
   - Admin has a right to delete all comment and add a comment for security purpose

2. recipe Management:
   - Display a list of recipe and their details, including cooking time, preparation time and serving.
   - Provide information about ingredients within each course, including diet.

3. comment Management:
   -provides a user to comment and review a recipe

4. Admin rights:
   - Right to manage comment section and secure it from negative comment.

ER Diagram:



Here in the above diagram N means many and double line means total participation.

-Here 1 means one and single line means partial participation.

Tables in SQL:

1. Recipe Table:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	RACIPE_ID	int			No	None		AUTO_INCREMENT	Change Drop More
2	RACIPE_NAME	varchar(200)	utf8mb4_0900_ai_ci		Yes	NULL			Change Drop More
3	PREP_TIME	int			No	None			Change Drop More
4	COOK_TIME	int			No	None			Change Drop More
5	SERVING	int			No	None			Change Drop More

2. About recipe Table:

Table structure | Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	RACIPE_ID	int			No	None			Change Drop More
2	DIET	varchar(100)	utf8mb4_0900_ai_ci		Yes	NULL			Change Drop More
3	COURSE	varchar(100)	utf8mb4_0900_ai_ci		Yes	NULL			Change Drop More
4	CUSINE	varchar(100)	utf8mb4_0900_ai_ci		Yes	NULL			Change Drop More

Check all With selected: Browse Change Drop Primary Unique Index Spatial

3. Ingredient details Table:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	INGREDIANT_ID	int			No	None			Change Drop More
2	INGREDIANT_NAME	varchar(100)	utf8mb4_0900_ai_ci		Yes	NULL			Change Drop More

4. Ingredient with recipe Table:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	racipe_id	int			Yes	NULL			Change Drop More
2	ingrediant_id	int			Yes	NULL			Change Drop More
3	quantity	varchar(450)	utf8mb4_0900_ai_ci		Yes	NULL			Change Drop More
4	remark	varchar(450)	utf8mb4_0900_ai_ci		Yes	NULL			Change Drop More

5. Login table:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	USER_ID	int			No	None			Change Drop More
2	USER_NAME	varchar(100)	utf8mb4_0900_ai_ci		No	None			Change Drop More
3	USER_TYPE	varchar(50)	utf8mb4_0900_ai_ci		Yes	NULL			Change Drop More
4	PASSWORD	varchar(400)	utf8mb4_0900_ai_ci		No	None			Change Drop More

6. Instruction table:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	RACIPE_ID	int			No	None			Change Drop More
2	INSTRUCTION	text	utf8mb4_0900_ai_ci		Yes	NULL			Change Drop More
3	URL	varchar(500)	utf8mb4_0900_ai_ci		Yes	NULL			Change Drop More

7.Comments table:

Table structure | Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	Cid	int			No	None		AUTO_INCREMENT	Change Drop More
2	RACIPE_ID	int			Yes	NULL			Change Drop More
3	msg	varchar(100)	utf8mb4_0900_ai_ci		Yes	NULL			Change Drop More
4	USER_ID	int			Yes	NULL			Change Drop More

Normal forms of the tables:

All tables are already in Third Normal Form (3NF), which means they have been normalized to remove any potential data redundancies and update anomalies. This normalization ensures data integrity and reduces the risk of inconsistencies in the database.

1. Recipe:

•First Normal Form (1NF): Appears to be in 1NF since each column contains atomic values, and there are no repeating groups.
•Second Normal Form (2NF): recipe_name uniquely identifies each row, and prep_time, cook_time, and serving depend only on recipe_name, this table is in 2NF.
•Third Normal Form (3NF): there are no transitive dependencies, this table would be in 3NF.

2. About_recipe:

•1NF: Appears to be in 1NF.
•2NF: recipe_id is a unique identifier and the other attributes depend solely on recipe_id, this table is in 2NF.
•3NF: there are no transitive dependencies among non-prime attributes, it should be in 3NF.

3. Ingredient_details:

•1NF: It's already in 1NF.
•2NF: ingredient_id is unique and ingredient_name depends solely on ingredient_id, it's in 2NF.
•3NF: It should be in 3NF because there are no transitive dependencies.

4. Ingredient_with_recipe:

•1NF: Appears to be in 1NF.
•2NF ingredient, recipe_id is unique and ingredient name depends solely on ingredient, it's in 2NF.
•3NF: there are no transitive dependencies among non-prime attributes, it should be in 3NF.

5. Login:

•1NF: it is in 1-NF
•2NF: Assuming user_id is a unique identifier and other attributes depend solely on user_id, it's in 2NF.
•3NF: It should be in 3NF if there are no transitive dependencies among non-prime attributes.

6. Instruction:

•1NF: Appears to be in 1NF.
•2NF: recipe_id uniquely identifies each row, and instruction and url depend only on recipe_id, this table is in 2NF.
•3NF: there are no transitive dependencies among non-prime attributes, it should be in 3NF.

7. Comments:

•1NF: Appears to be in 1NF.
•2NF: cid is a unique identifier and other attributes depend solely on cid, it's in 2NF.
•3NF: It should be in 3NF because there are no transitive dependencies among non-prime attributes.

All used Sql queries:

- **register**

$sql = "INSERT INTO `log_in`(`USER_NAME`, `email`, `PASSWORD`) VALUES('$name', '$email', '$password')";

$sql = "SELECT * from log_in where email='$email' and password='$password' order by USER_ID desc limit 1";

- **login**

$sql = "SELECT * from log_in where email='$email' and password='$password' order by USER_ID desc limit 1";

- **commentid**

$sql = "INSERT INTO `comments`(`RACIPE_ID`, `msg`, `USER_ID`) VALUES ('$id','$msg','$uid')";
$sql = "DELETE FROM `comments` WHERE cid='$cid'";
$sql = "UPDATE `comments` SET msg='$msg' WHERE cid='$cid'";

- **numbers**

$sql = "SELECT count(*) as rs FROM racipe";
$sql = "SELECT count(*) as rs FROM ingrediant_details";
$sql = "SELECT count(DISTINCT cusine) as rs FROM about_recipe";
$sql = "SELECT count(DISTINCT diet) as rs FROM about_recipe";

- **Select queries for search in random**

$sql = "SELECT r.racipe_id as id,a.cusine as cu, a.diet as d, a.course as co, r.RACIPE_NAME as rname, r.prep_time as pt, r.cook_time as ct, r.serving as ser FROM about_recipe a join racipe r on a.racipe_id=r.racipe_id WHERE r.racipe_id in (5,500,456,789,451,2000,985,6544,5800) limit 9";

$sql = "SELECT r.racipe_id as id,a.cusine as cu, a.diet as d, a.course as co, r.RACIPE_NAME as rname, r.prep_time as pt, r.cook_time as ct, r.serving as ser FROM about_recipe a join racipe r on a.racipe_id=r.racipe_id order by r.racipe_id desc limit 9";

- **recipe specific**

$sql = "SELECT r.racipe_id as id,a.cusine as cu, a.diet as d, a.course as co, r.RACIPE_NAME as rname, r.prep_time as pt, r.cook_time as ct, r.serving as ser FROM about_recipe a join racipe r on a.racipe_id=r.racipe_id WHERE r.RACIPE_ID = '$id'";

- **ingrediant**

$sql = "SELECT r.quantity as q,r.remark as r,a.INGREDIANT_NAME as n FROM ingrediant_details a join ingrediant_for_racipe r on r.ingrediant_id = a.ingrediant_id where r.racipe_id = '$id'" ;

--instructions

$sql = "SELECT * FROM `instructions` where RACIPE_ID='$id'" ;

--comets

$sql = "SELECT a.USER_ID as uid,a.msg as msg,b.user_name as name,a.Cid as Cid FROM comments a join log_in b on a.USER_ID=b.USER_ID WHERE RACIPE_ID=$id";

- **racipe search**

-- searched all number only

$sql ="SELECT count(*) as rs FROM about_recipe a join racipe r on a.racipe_id=r.racipe_id WHERE (cusine = '$cuisine' OR 'All' = '$cuisine') AND (diet = '$diet' OR 'All' = '$diet') AND (course = '$course' OR 'All' = '$course') AND (RACIPE_NAME like '%$name%' or 'All' = '$name') AND (PREP_TIME+COOK_TIME < $time) ";