|  |
| --- |
| **HOUSE KEEPING SYSTEM**  **18CSC209J - Database Management System and Cloud Integration Services**  **Mini Project Report**  *Submitted by*  **SHRAJOY SUR [Reg. No.: RA2111028010182]**  **B.Tech. CSE – CLOUD COMPUTING**  **SRMIST-01.jpg**  **DEPARTMENT OF NETWORKING AND COMMUNICATION**  **SCHOOL OF COMPUTING**  **COLLEGE OF ENGINEERING AND TECHNOLOGY**  **SRM INSTITUTE OF SCIENCE AND TECHNOLOGY**  **(Under Section 3 of UGC Act, 1956)**  S.R.M. NAGAR, KATTANKULATHUR – 603 203  KANCHEEPURAM DISTRICT  **MAY 2023** |

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **Chapter No.** | **Title** | **Page No.** |
| 1. | Abstract |  |
| 2. | Introduction |  |
| 3. | Problem Statement |  |
| 4. | Module Description |  |
| 5. | Use case diagram |  |
| 6. | ER diagram |  |
| 7. | Database Creation using DDL and DML |  |
| 8. | Normalization of Database |  |
| 9. | Implementation using Dynamo DB |  |
| 10. | Conclusion |  |
| 11 | Appendix I – Screenshot |  |
| 12 | Appemdix II - Github Profile |  |
| 13 | Appendix III - AWS Course Completion certificate |  |

**Abstract**

Today the world is moving towards automation and simplification of various processes that generally needed human intervention and decision making with the help of the advanced technology that we have at our disposal today. One such attempt to automate tedious processes is the advent of “On Demand Service applications” these applications are used by the user to avail services as and when need arise, this concept is getting increasingly popular and has found its use in many service domains, such as hotels booking, food ordering, taxi booking etc. The proposed system is also an on demand application for availing housing services by the users. This will significantly reduce the time and effort required to scout and hire professionals who provides services such as cooking, cleaning, babysitting and other such services required in daily life. As in many on demand applications the goal will be to provide the required service on the user’s demand.

Keywords: UHS (Urban Housekeeping Services), On Demand Application, Housekeeping, Maids, Hiring Housekeepers

**Introduction**

HOUSEKEEPING SYSTEM is a project to automate the “Process of scouting and hiring professionals who provide housekeeping services” by developing a system that is efficient and user-friendly. The main idea of the project is to maintain and handle all domains involved, from cooking, cleaning to babysitting and many other domains. There is a need to automate this sector because till now it is managed manually. So, by making a proper system for this process, the overall process can be done efficiently and proper utilization of resources can be done. It will also aim at providing a clear picture of how the entire process is taking place stepwise so that the chances of frauds and errors can be minimized. UHS will fulfil the purpose of providing easy accessibility to its customers.

In this project, there are many domains included in it, so customers basically have to choose his/her domain in which he/she is interest to get the service and after that we will show a list of housekeepers on the basis of availability and rating (high to low) to the customers and after his/her interest he/she select one of them.

After this we will allocate the housekeeper to the customer. In this project, there is also a feature of the ratings for the customer which will be assigned by us depending on the housekeeper’s feedback.

There is a feature of housekeepers’ safety and also there are measures for customer security and hygiene. UHS is the need of the hour in today’s rapidly growing world. Hence, this project will prove to be a boon to the housekeeping domain by the easy accessibility of all the listed services. Also, this will increase employment opportunity within the country.

**Problem Statement**

The Problem that this project tries to tackle is the amount of time and effort that people have to invest in order to hire individuals for housekeeping, this is especially inconvenient in the urban areas because the whole point of hiring someone to take care of their house is to save their own time as they are already preoccupied in their day jobs or other activities and are unable to perform these tasks such as cooking, cleaning, babysitting etc.

The objective of the Housekeeping System is to develop a database for website so that the users can interact with the system and as per their needs they can hire professionals to avail the services provided by them

**Module Description**

This project provides a system which can be utilized by the parties interested in hiring professionals for various housekeeping activities and other miscellaneous day to day activities by listing all the available service providers nearby and allowing the user to select one that suits their interest.

The platform will remove the complexities of legal background checks for the user and make it abstract i.e. the user will not have to worry about doing any background probes on the professionals that they are hiring as all the profiles that appears on the application will be pre-verified.

The platform will also integrate with a payment gateway to enable users to make safe and fast payments on the go. The application will also provide the user with an option to cancel their request with ease. The system consists of two actors:

1. User

2. Admin

User: - Is the one who uses the application for hiring purpose. User can refer the housekeeper too.

Admin: - Is the one who manages the whole system, basically the admin manage the whole database of the housekeepers, and also deals with the customer requests and complaints. Admin will also verify the profiles of the housekeepers and then add them the database

**Use case diagram**

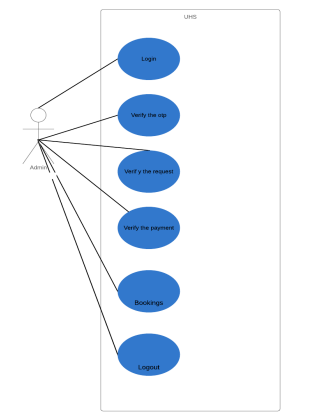
Use case diagram is a behavioral UML diagram type and frequently used to analyze various systems. They enable you to visualize the different types of roles in a system and how those roles interact with the system.

* **To identify functions and how roles interact with them** – The primary purpose of use case diagrams.
* **For a high-level view of the system** – Especially useful when presenting to managers or stakeholders. You can highlight the roles that interact with the system and the functionality provided by the system without going deep into inner workings of the system.
* **To identify internal and external factors** – This might sound simple but in large complex projects a system can be identified as an external role in another use case.

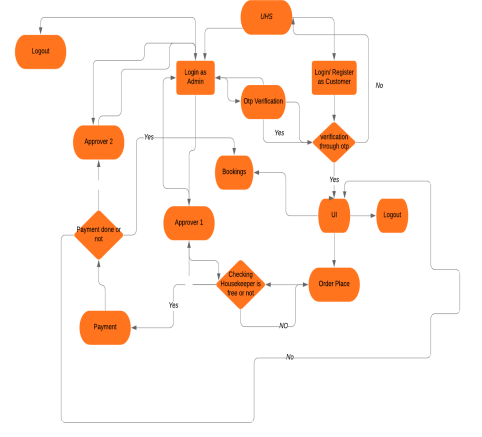
USECASE DIGRAM FOR OUR PROJECT WILL BE:-



* USECASE DIGRAM FOR USER



* Usecase digram for admin

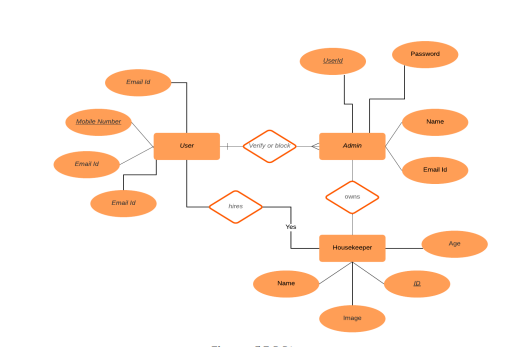


**SYSTEM FLOWCHART DIGRAM**

**ER diagram**

The Entity Relational Model is a model for identifying entities to be represented in the database and representation of how those entities are related. The ER data model specifies enterprise schema that represents the overall logical structure of a database graphically.

The Entity Relationship Diagram explains the relationship among the entities present in the database. ER models are used to model real-world objects like a person, a car, or a company and the relation between these real-world objects. In short, ER Diagram is the structural format of the database.



**Database Creation using DDL and DML**

DATABASE USED:- SQLPlus

---------------------------------------------Sql commands-----------------------------------------

SQL> CREATE TABLE userdetails(u\_id int,U\_name varchar2(20),mobile\_no int,email varchar2(30));

Table created.

SQL> desc userdetails;

Name Null? Type

----------------------------------------- -------- ----------------------------

U\_ID NUMBER(38)

U\_NAME VARCHAR2(20)

MOBILE\_NO NUMBER(38)

EMAIL VARCHAR2(30)

SQL> CREATE TABLE admin(userid int,User\_name varchar2(20),password varchar2(5),email varchar2(30));

Table created.

SQL> desc admin;

Name Null? Type

----------------------------------------- -------- ----------------------------

USERID NUMBER(38)

USER\_NAME VARCHAR2(20)

PASSWORD VARCHAR2(5)

EMAIL VARCHAR2(30)

SQL> CREATE TABLE housekeeper(emp\_id int,keepername varchar2(20),service varchar2(5),service\_id int,age int)

Table created.

SQL> desc housekeeper;

Name Null? Type

----------------------------------------- -------- ----------------------------

EMP\_ID NUMBER(38)

KEEPERNAME VARCHAR2(20)

SERVICE VARCHAR2(5)

SERVICE\_ID NUMBER(38)

AGE NUMBER(38)

SQL> CREATE TABLE service(u\_id int,s\_id int,keeper\_id int,address varchar2(30));

Table created.

SQL> desc service;

Name Null? Type

----------------------------------------- -------- ----------------------------

U\_ID NUMBER(38)

S\_ID NUMBER(38)

KEEPER\_ID NUMBER(38)

ADDRESS VARCHAR2(30)

SQL> INSERT INTO userdetails VALUES(001,'ARIN',987612345,'ar@gmail');

1 row created.

SQL> INSERT INTO userdetails VALUES(010,'DEEPAK',432612345,'dp@gmail');

1 row created.

SQL> INSERT INTO userdetails VALUES(020,'SANDEEP',902612345,'snd@gmail');

1 row created.

SQL> INSERT INTO userdetails VALUES(030,'GOVIND',7026717845,'gv@gmail');

1 row created.

SQL> INSERT INTO userdetails VALUES(108,'RUPALI',6028717845,'ru@gmail');

1 row created.

SQL> SELECT \* FROM userdetails

U\_ID U\_NAME MOBILE\_NO EMAIL

---------- -------------------- ---------- ------------------------------

1 ARIN 987612345 ar@gmail

10 DEEPAK 432612345 dp@gmail

20 SANDEEP 902612345 snd@gmail

30 GOVIND 7026717845 gv@gmail

108 RUPALI 6028717845 ru@gmail

SQL> INSERT INTO admin VALUES(1001,'SHRAJOY','sh123','shrajoy@gmail');

1 row created.

SQL> INSERT INTO admin VALUES(2002,'AYUSH','ay123','ayush@gmail');

1 row created.

SQL> INSERT INTO admin VALUES(3003,'MANAS','ma123','manas@gmail');

1 row created.

SQL> SELECT \* FROM admin;

USERID USER\_NAME PASSW EMAIL

---------- -------------------- ----- ------------------------------

1001 SHRAJOY sh123 shrajoy@gmail

2002 AYUSH ay123 ayush@gmail

3003 MANAS ma123 manas@gmail

SQL> INSERT INTO housekeeper VALUES(01,'RAJANEESH','MAID',110,19);

1 row created.

SQL> INSERT INTO housekeeper VALUES(02,'SHIVA','COOK',220,10);

1 row created.

SQL> INSERT INTO housekeeper VALUES(03,'RAVI','GUARD',330,30);

1 row created.

SQL> INSERT INTO housekeeper VALUES(04,'RAMU','MID',110,33);

1 row created.

SQL> INSERT INTO housekeeper VALUES(04,'ARVIND','MAID',110,32);

1 row created.

SQL> DELETE FROM housekeeper where KEEPERNAME='RAMU';

1 row deleted.

SQL> DELETE FROM housekeeper where KEEPERNAME='ARVIND';

1 row deleted.

SQL> INSERT INTO housekeeper VALUES(05,'RAMU','MAID',110,34);

1 row created.

SQL> INSERT INTO housekeeper VALUES(04,'ARVIND','MAID',110,32);

1 row created.

SQL> INSERT INTO housekeeper VALUES(06,'SHIVANI','COOK',220,22);

1 row created.

SQL> INSERT INTO housekeeper VALUES(07,'SAHIL','GUARD',330,45);

1 row created.

SQL> SELECT \* FROM HOUSEKEEPER;

EMP\_ID KEEPERNAME SERVI SERVICE\_ID AGE

---------- -------------------- ----- ---------- ----------

1 RAJANEESH MAID 110 19

2 SHIVA COOK 220 10

3 RAVI GUARD 330 30

5 RAMU MAID 110 34

4 ARVIND MAID 110 32

6 SHIVANI COOK 220 22

7 SAHIL GUARD 330 45

7 rows selected.

SQL> INSERT INTO service VALUES(001,110,01,'juhu');

1 row created

SQL> INSERT INTO service VALUES(020,220,02,'chrompet');

1 row created.

SQL> INSERT INTO service VALUES(108,330,07,'egmore');

1 row created.

SQL> INSERT INTO service VALUES(108,110,05,'egmore');

1 row created.

SQL> SELECT \* FROM service;

U\_ID S\_ID KEEPER\_ID ADDRESS

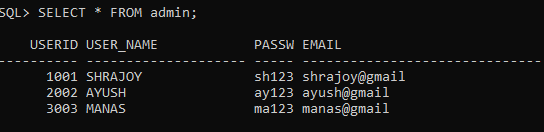
---------- ---------- ---------- ------------------------------

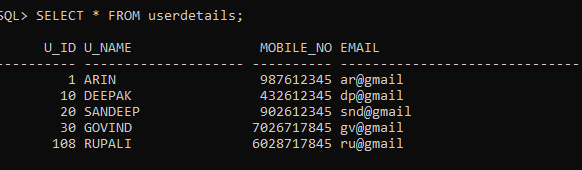
1 110 1 juhu

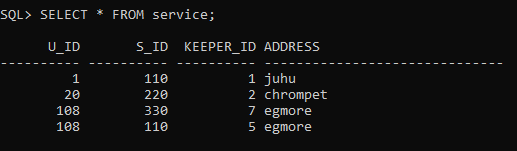
20 220 2 chrompet

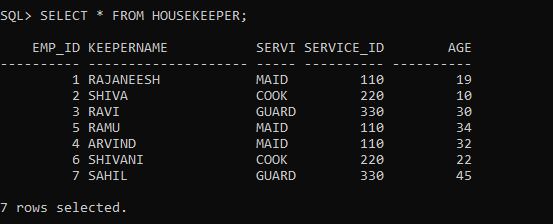
108 330 7 egmore

108 110 5 egmore









**Normalization of Database**

* Normalization is the process of organizing the data in the database.
* Normalization is used to minimize the redundancy from a relation or set of relations. It is also used to eliminate undesirable characteristics like Insertion, Update, and Deletion Anomalies.
* Normalization divides the larger table into smaller and links them using relationships.
* The normal form is used to reduce redundancy from the database table.

The main reason for normalizing the relations is removing these anomalies. Failure to eliminate anomalies leads to data redundancy and can cause data integrity and other problems as the database grows. Normalization consists of a series of guidelines that helps to guide you in creating a good database structure.

|  |  |
| --- | --- |
| **Normal Form** | **Description** |
| 1NF | A relation is in 1NF if it contains an atomic value. |
| 2NF | A relation will be in 2NF if it is in 1NF and all non-key attributes are fully functional dependent on the primary key. |
| 3NF | A relation will be in 3NF if it is in 2NF and no transition dependency exists. |
| BCNF | A stronger definition of 3NF is known as Boyce Codd's normal form. |
| 4NF | A relation will be in 4NF if it is in Boyce Codd's normal form and has no multi-valued dependency. |
| 5NF | A relation is in 5NF. If it is in 4NF and does not contain any join dependency, joining should be lossless. |

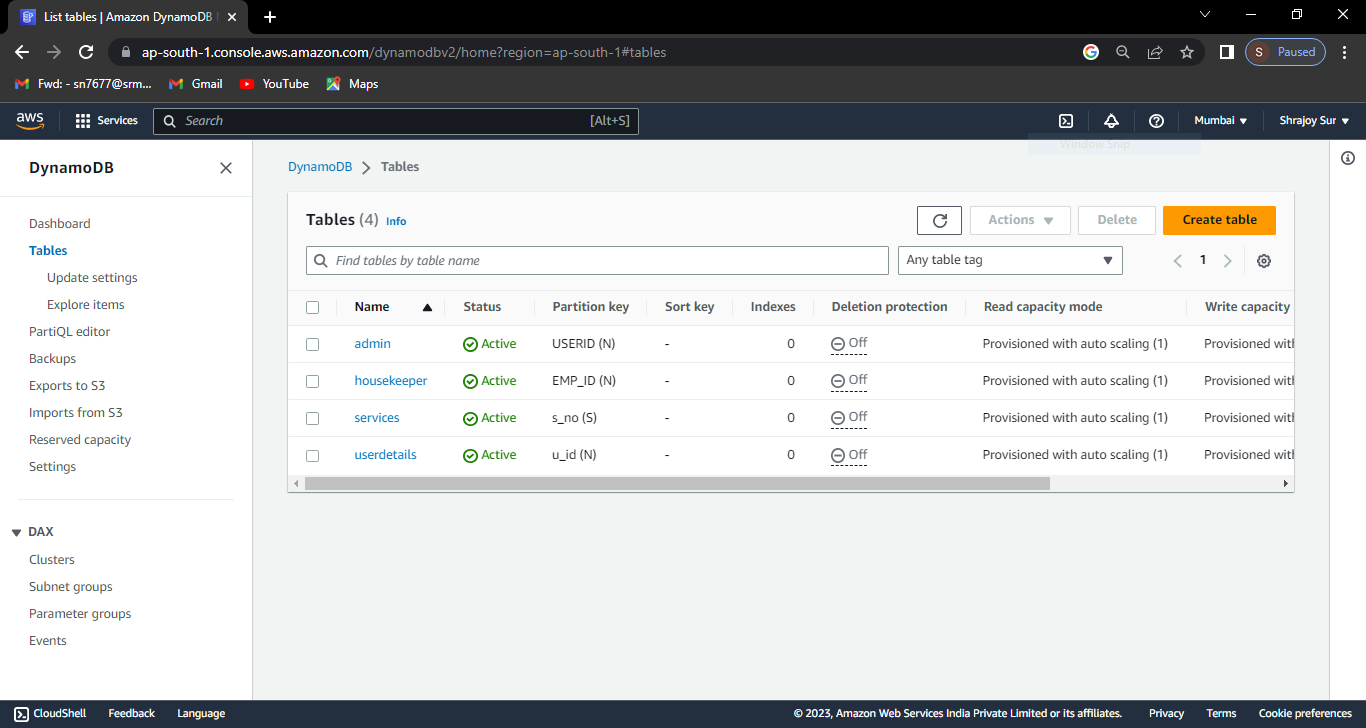
\*\*FOR OUR CURRENT DATABASE THERE IS NO SUCH NEED FOR NORMALISATION .AT MAX WE CAN DO IS 1NF NORMALIZATION.

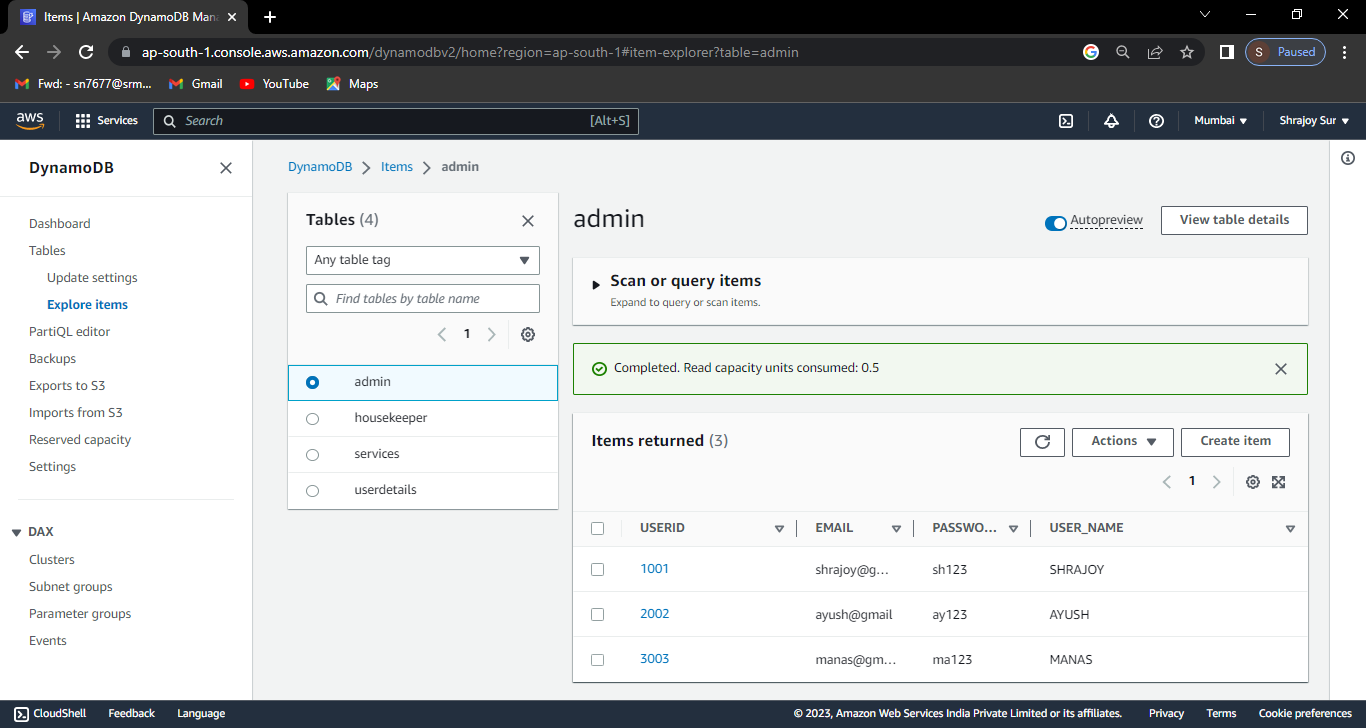
BUT IN FUTURE THE AS THE DATASET INCREASES WE MIGHT NEED THE NORMALIZATION PLANS.\*\*

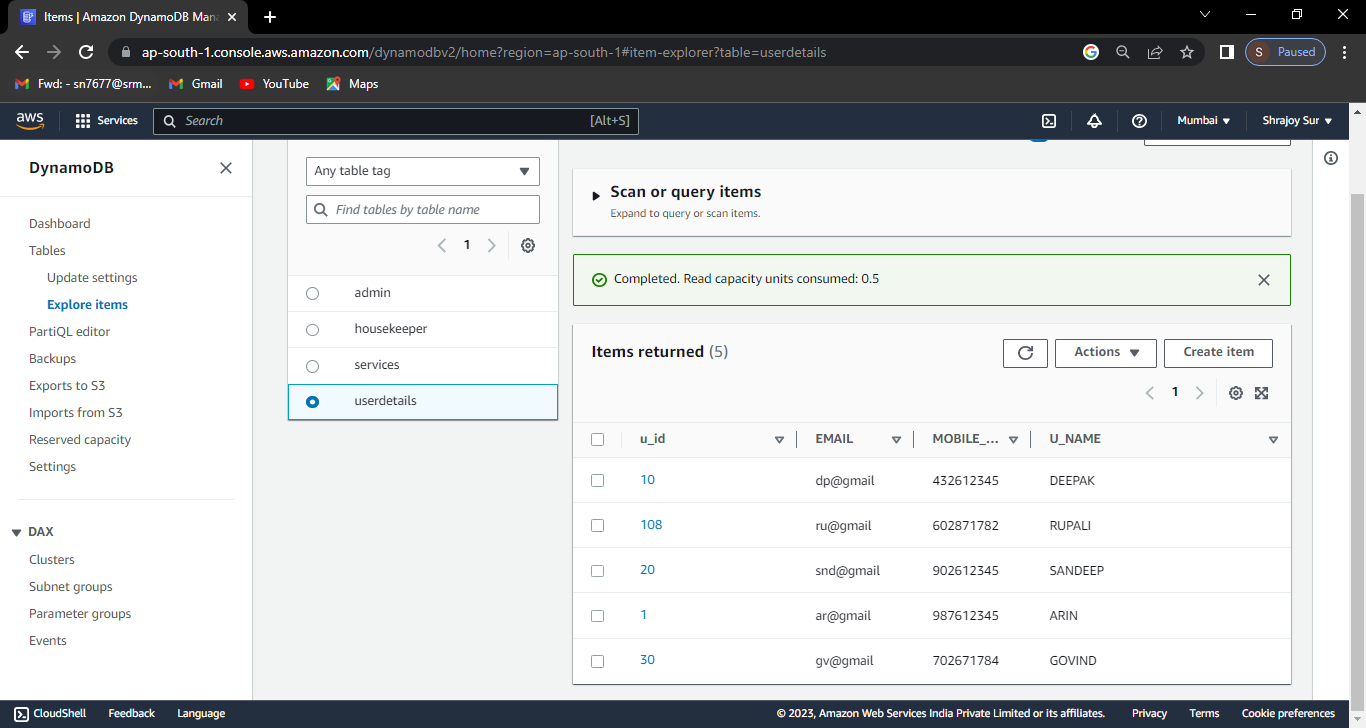
**Implementation using Dynamo DB**

Amazon DynamoDB is a fully managed, serverless, key-value NoSQL database designed to run high-performance applications at any scale. DynamoDB offers built-in security, continuous backups, automated multi-Region replication, in-memory caching, and data import and export tools.

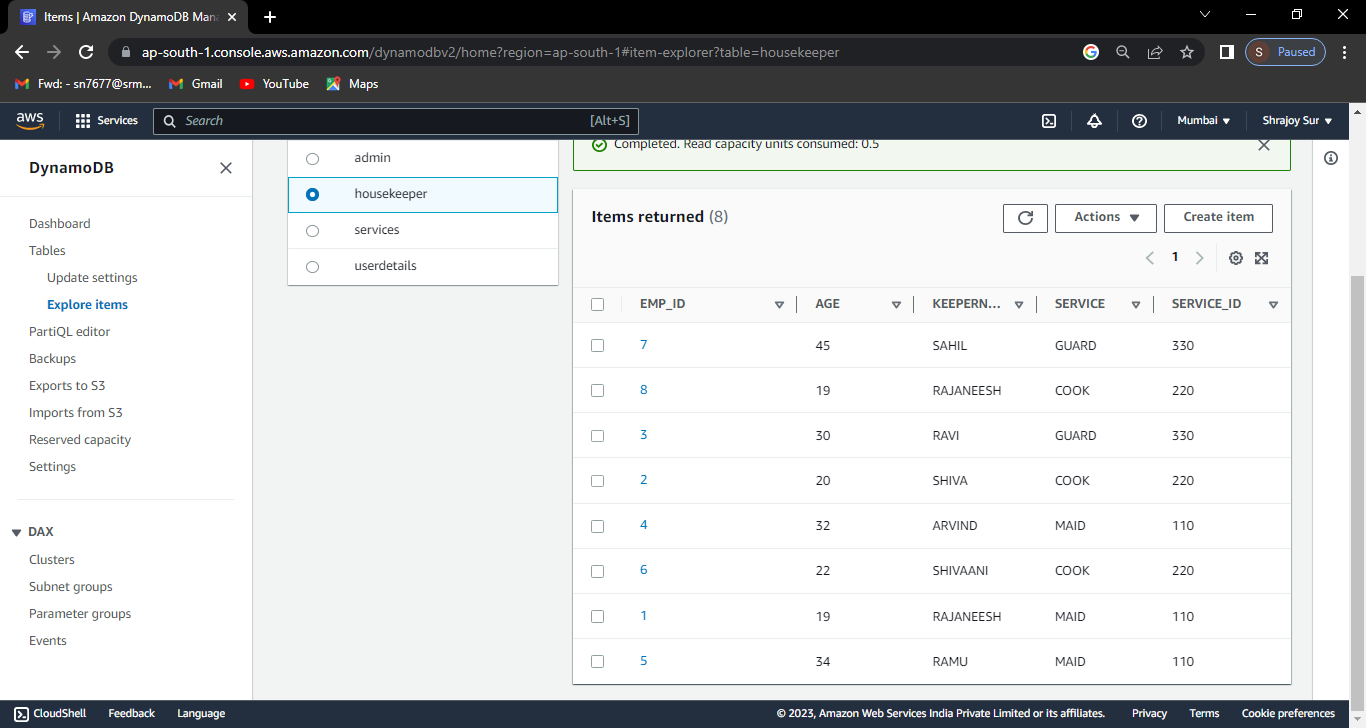
For HOUSEKEEPING SYSTEM we have created NOSQL database in DynamoDB.

 \*\*all the tables in the DynamoDB.\*\*

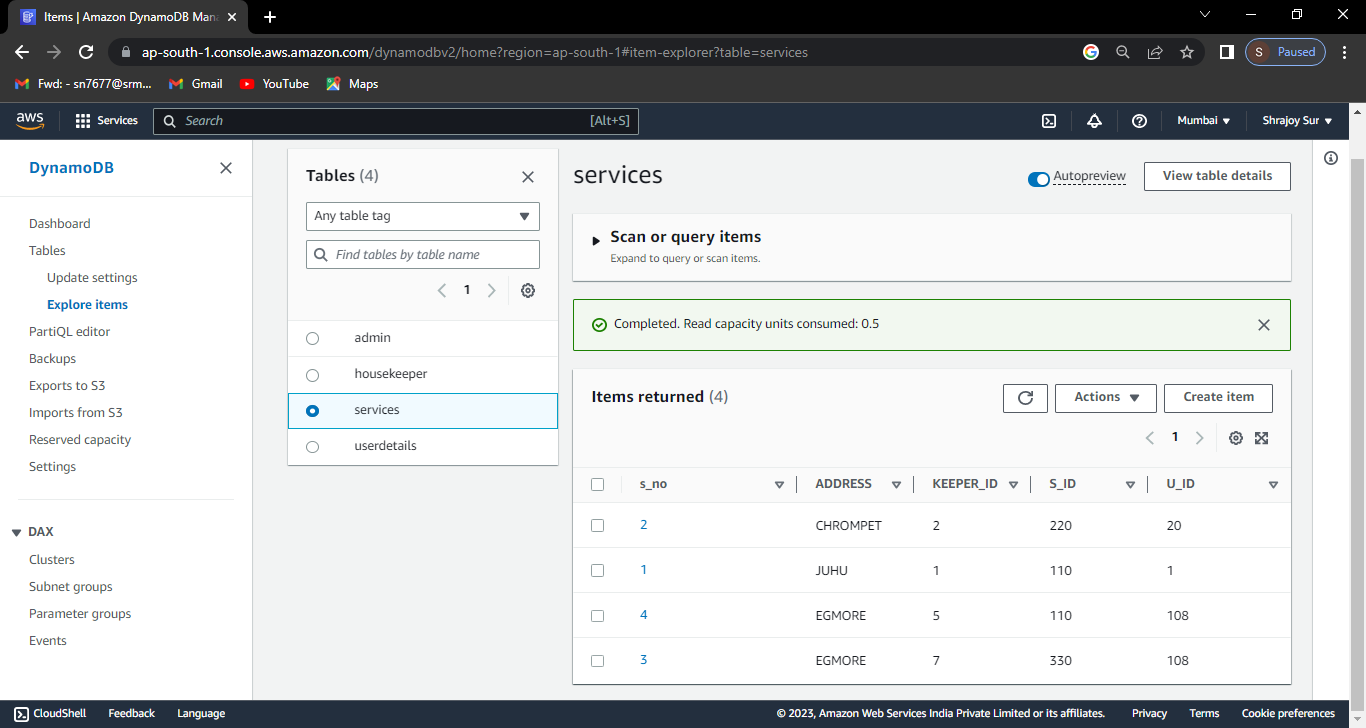
\*\*admin table items\*\*



\*\*userdetails table items\*\*



\*\*housekeeper table items\*\*



\*\*service table items\*\*

**Conclusion**

Housekeeping System application will be able to solve various problems faced by the people working day jobs or involved in other activities due to which are unable to spare time for scouting and hiring housekeepers by providing a platform for the same.

This application will also help in ensuring user satisfaction and reduce the stress of the professionals who get hired through this platform and provide them with a platform which will significantly increase their chances of getting hired.