FACULTY OF COMPUTING
UTM Johor Bahru

SESSION 2023/2024 SEMESTER 1

SECD2523 - DATABASE

PROJECT PHASE III

Group Name : Agent P
Course : COMPUTER NETWORKS & SECURITY
Section : 02
Title : Talent System
Task : Database Logical Design & SQL
Lecturer : Dr. Izyan Izzati binti Kamsani

GROUP MEMBERS:

No.	Name	Matric No.
1	NAVINDRAN A/L RAGHUPATHY	A22EC0227
2	KUGANRAJ A/L RAMESH	A22EC0177
3	KUGHANRAJ A/L ARUNASALAM	A22EC0179
4	KUGANES VARMAN A/L BALAMN	A22EC0176

Table Of Contents

1.0 Introduction

In the 21st century, writing a good CV is very important for current fresh graduates and job seekers since most companies now use ATV to track CVs. GetMe Technology PLT provides the best solution for this problem.GetMe Technology PLT is a business that has been around since 2019.This company is the creator and also the owner of the website "GetMe Hired.io," which generates and evaluates CVs for its clients based on the package to which the customers have subscribed.However ,it is difficult for the GetMe Technology PLT company to provide the highest quality services to their customers because of the challenges that they face. The lack of a proper system for companies that are employing workers in their company and also for proofreaders who evaluate clients' CVs is one of the primary challenges faced by the GetMe Technology PLT. It makes things more difficult for the clients, as well as for the companies who collaborate with GetMe Technology PLT in some way.Proofreaders also face some difficulties since they will receive the CVs filled by the customers via an email platform which is not efficient. Managing CVs in using an email inbox can become very disorganized, especially when the number of CVs increases. GetMe Technology PLT is currently using WhatsApp manually as their primary communication channel in order to respond to questions from consumers and also to resolve issues that customers have had with the payment process. WhatsApp might be suitable for small-scale customers, but it would become a major problem if it came to managing a large number of customers. Because of this, some of the customers may experience a delay in order to get a response.

Thus to address the problem, a new system for GetMe Technology PLT was proposed. In this new system there will be Automated CV Template Forwarding which after the customer makes the payment, a system will automatically send the CV templates chosen by the customer to their email that they have filled earlier. Secondly, there will be a platform for the admins to share information on a regular basis.Thus, admins will be able to communicate between themselves and have an effective information sharing platform. This will help both the customers and the sales support team as any sales support can help any customer at any given time rather than only one sales support knowing the problem of the specific customer. Furthermore, there also will be a platform for communication between the admins and users. With a platform specifically made for this communication, the customer may put more attention to the notification of the platform and prioritize the messages sent.In the new

system, there will be a library system for the companies in collaboration with to headhunt their preferred candidate. This library system will provide the company's recruiter with the customer's resume and the skills they have. This will make it easier for the company to find the person most suitable to them. Customer data will always be available in the library so even if they have landed a job, the recruiter can anytime message the customer and they can discuss it between themselves. Finally, there will be all consultants to consult users if they have a problem with their CV. When customer send out their filled up CV template and the proofreader finds some problems in their CV, customers can receive a consultation to solve the problem and generate a full CV. Hence, with all the new features, the system can work much better than the current system and satisfy the customer more.

3.0 Data & Transaction requirement

3.1 Proposed business rules

1) Each customer has zero or more CV
2) Each company can view at least one CV
3) Each CV can be viewed by one or many companies
4) Each company hire at least one customers
5) Each customer will be hired by only one company
6) Each consultant consult at least one customer
7) Each customer consulted by only one consultant
8) Each customer can ask zero or more questions
9) Question can be asked by one or many customers
10) Each admin answer at least one question
11) Each questions answered by at least one admin
12) One admin will assign one or many proofreader
13) Each proofreader will assigned by only one admin
14) Each proofreader generate one or more CV
15) CV generated by only one proofreader

3.2 Proposed data & transactional

Data requirement

1) **Customer**

   Customers will select and purchase a package to get a CV template. The information of users such as name, contact number, ID, email are stored as data. Each customer has their own username and password.

2) **CV**

   Each CV has its own ID, the customer's ID , and the proofreader's ID. Then, the description of the CV and the skills that customers have.

3) **Company**

   These companies have their name, ID, contact number, email, address. Each company has their username and password. Then, the jobs that are provided by the company.

4) **Consultant**

   The consultant has their ID and password to login to the system. Each consultant has a name, email, and address. The qualification and availability will be given by the consultant to the customer to know more about them.

5) **Q&A**

   Questions and answers will be asked by the customer and answered by the admin which contains questions' ID and answers' ID and their description respectively.

6) **Admin**

   Each admin has their name, contact number, email, address, position. They have their ID and password to login to the system.

7) **Proofreader**

   Each proofreader has a name, contact number, email, address, availability time to give to the customer. They have an ID and password to login as usual. Proofreaders have the CV ID that they generate to customers.

Transaction Requirement

Data Entry

- Enter the details of the customers
- Select the package for the CV
- Enter the details of the company
- Enter the details of the consultant
- Enter the details of the Admin
- Enter the details of the proofreader
- Enter the question lists
- Enter the answers for the questions
- Enter the details for the CV

Data Update / Deletion

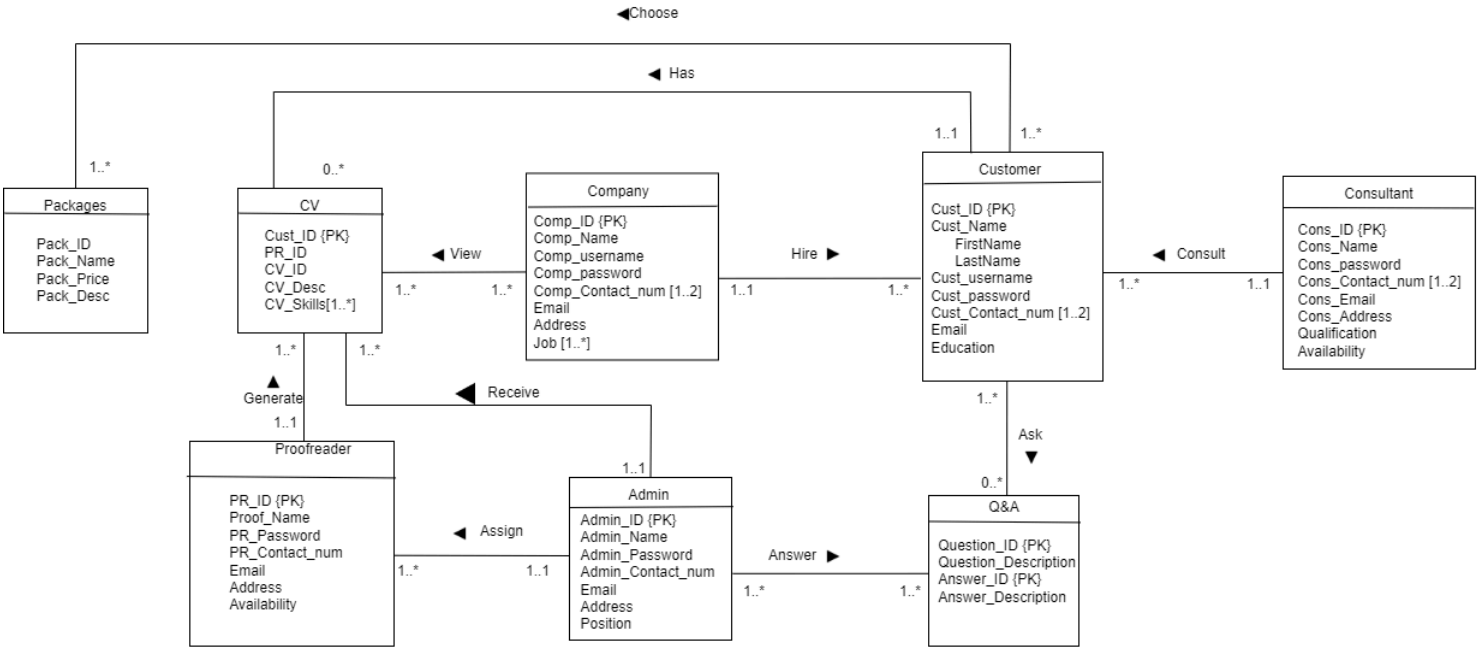
- Update / deletion the details of the customers
- Update / deletion the details of the company
- Update / deletion the details of the consultant
- Update / deletion the details of the admin
- Update / deletion the details of the proofreader
- Update / deletion of the question and answers
- Update / deletion of the question and CV

Data Query

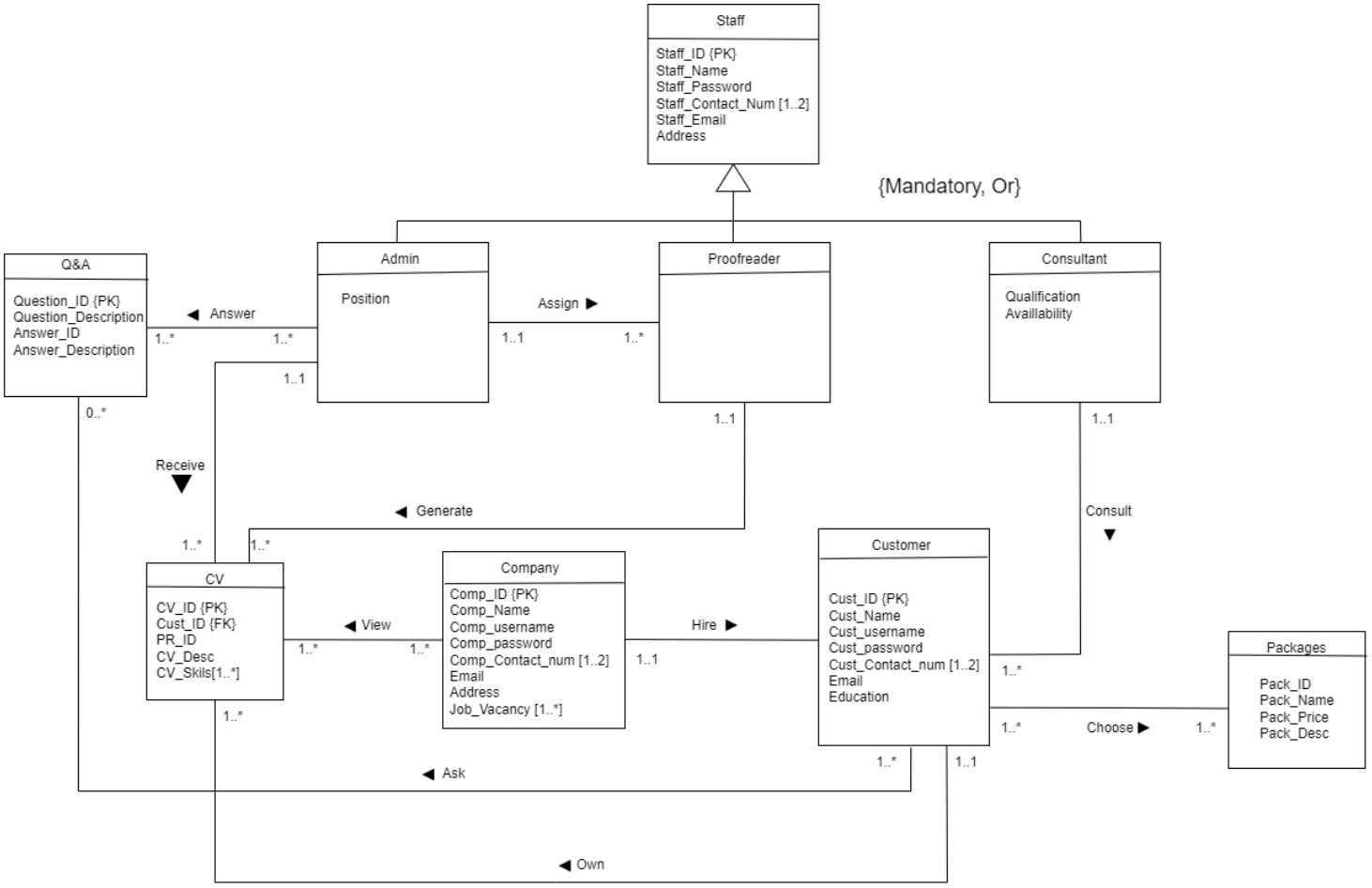
- List the packages of the CV for customers
- List the questions asked by customers
- List the answers replied by admin

4.0 Database Conceptual Design

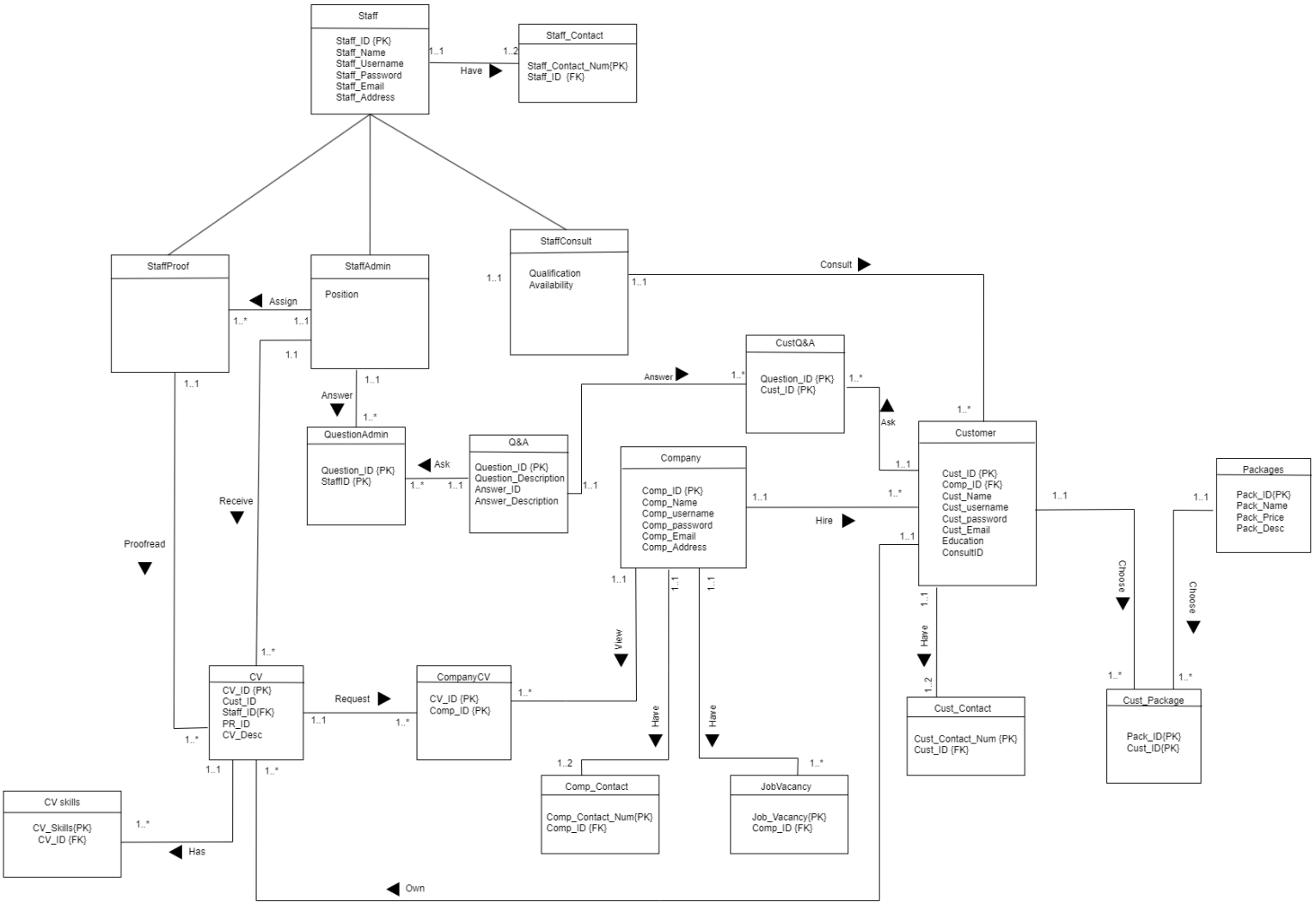
4.1 Conceptual ERD



4.2 Enhanced ERD



4.3 Logical EERD



5.0 Data dictionary

Entity Name	Attributes	Description	Data type & Length	Nullity
CUser	Cust_username {PK} Cust_password	Username of customer for authentication Password of customer	varchar2(50) varchar2(50)	No No
CEmail	Cust_Email {PK} Cust_Username	Email of the customer Username of customer for authentication	varchar2(50) varchar2(50)	No No
CID	Cust_ID {PK} Cust_Name Cust_Email Education Comp_ID ConsultID	Unique ID for customer Name of the customer Email of the customer Highest education level of the customer Unique ID for company Unique ID for consultant	varchar2(5) varchar2(50) varchar2(50) varchar2(50) varchar2(5) varchar2(50	No No No No No No
Customer	Cust_ID {PK} Cust_Email {PK}	Username of customer for authentication Email of the customer	varchar2(5) varchar2(50)	No No
CoUser	Comp_username {PK} Comp_password	Username of company for authentication Password of company	varchar2(50) varchar2(50)	No No
CoID	Comp_ID {PK} Comp_Name Comp_Email Address	Unique ID for company Name of the company Email of the company Address of the company	varchar2(5) varchar2(50) varchar2(50) varchar2(50)	No No No No
CoEmail	Comp_Email {PK} Comp_username	Email of the company Username of company for authentication	varchar2(50) varchar2(50)	No No
Company	Comp_ID {PK} Comp_Email {PK}	Unique ID for company Email of the company	varchar2(5) varchar2(50)	No No
CV	CV_ID {PK} Cust_ID PR_ID CV_Desc	Unique ID of the CV's proofreader Unique ID of the CV Description of the CV	varchar2(20) varchar2(20) varchar2(5) varchar2(150)	No No No No

		Skills of the customer		
ConsultantUser	Consult_username {PK} Consult_Password	Username of consultant for authentication Password of consultant	varchar2(50) varchar2(5)	No No
ConsultantID	Consult_ID {PK} Consult_Name Address Qualification Availability	Unique ID for consultant Name of the consultant Address of the consultant Qualification held by the consultant Available time slot	varchar2(5) varchar2(50) varchar2(50) varchar2(50) Date	No No No No No
ConsultantEmail	Consult_Email {PK} Consult_username	Email of the consultant Username of consultant for authentication	varchar2(50) varchar2(50)	No No
StaffConsult	Consult_ID {PK} Consult_Email {PK}	Email of the consultant Username of consultant for authentication	varchar2(20) varchar2(50)	No No
AdminUser	Staff_Username {PK} Staff_Password	Username of staff for authentication Password of staff	varchar2(50) varchar2(5)	No No
AdminID	Staff_ID {PK} Staff_Name Staff_Address Position	Unique ID for staff Name of the staff Address of the staff Position held by staff	varchar2(5) varchar2(50) varchar2(50) varchar2(50)	No No No No
AdminEmail	Staff_Email {PK} Staff_Username	Email of the staff Username of staff for authentication	varchar2(50) varchar2(50)	No No
StaffAdmin	Staff_ID {PK} Staff_Email {PK}	Unique ID for staff Email of the staff	varchar2(5) varchar2(50)	No No
ProofUser	Staff_Username {PK} Staff_Password	Username of staff for authentication Password of staff	varchar2(50) varchar2(5	No No
ProofID	Staff_ID {PK} Staff_Name Staff_Address Position	Unique ID for staff Name of the staff Address of the staff Position held by staff	varchar2(5) varchar2(50) varchar2(50) varchar2(50)	No No No No
ProofEmail	Staff_Email {PK} Staff_Username	Email of the staff Username of staff for authentication	varchar2(50) varchar2(50)	No No

StaffProof	Staff_ID {PK} Staff_Email {PK}	Unique ID for staff Email of the staff	varchar2(5) varchar2(50)	No No
Question	Question_ID {PK} Question_Description	Unique ID for question Description for question	varchar2(5) varchar2(100)	No No
Answer	Answer_ID {PK} Answer_Description	Unique ID for answer Description for answer	varchar2(5) varchar2(100)	No No
Q&A	Question_ID {PK} Answer_ID {PK}	Unique ID for question Unique ID for answer	varchar2(5) varchar2(5)	No No
CustQ&A	Question_ID {PK} Cust_ID Answer_ID	Unique ID of question Unique ID for customer Unique ID for answer	varchar2(5) varchar2(5) varchar2(5)	No No
AdminQ&A	Question_ID {PK} Admin_ID Answer_ID	Unique ID of question Unique ID for admin Unique ID for answer	varchar2(5) varchar2(5) varchar2(5)	No No No
JobVacancy	Job_Vacancy {PK} Comp_ID	Jobs offered by the company Unique ID for company	varchar2(50) varchar2(5)	No No
Comp_Contact	Comp_Contact_Num Comp_ID	Phone number of company Unique ID for company	varchar2(50) varchar2(5)	No No
Cust_Contact	Cust_Contact_Num Cust_ID	Phone number of customer Unique ID for customer	varchar2(50) varchar2(5)	No No
AdminContact	Contact Staff_ID	Phone number of customer Unique ID for customer	varchar2(50) varchar2(5)	No No
ProofContact	Contact Staff_ID	Phone number of customer Unique ID for customer	varchar2(50) varchar2(5)	No No
ConsultContact	Contact Staff_ID	Phone number of customer Unique ID for customer	varchar2(50) varchar2(5)	No No
Package	Pack_ID Pack_Name Pack_Price	Unique ID for Package Name of the package Price for the package	varchar2(5) varchar2(50) number2 (4)	No No No

	Pack_Desc	Description of the package	varchar2(50)	No
Cust_Pack	Cust_ID Pack_ID	Unique ID for Customer Unique ID for Package	varchar2(5) varchar2(2)	No No

6.0 Normalization

6.1 Relation Schemas

1. **Customer**
   (Cust_ID,Cust_Name,Cust_username,Cust_password,Cust_Email,Education, Comp_ID,ConsultID)
   PK : Cust_ID
   FK : Comp_ID reference Company(Comp_ID)
   ConsultID reference StaffConsult(ConsultID)

2. **Company**(Comp_ID,Comp_Name,Comp_username,Comp_password, Comp_Email ,Address)
   PK : Comp_ID

3. **CV**(CV_ID ,Cust_ID,PR_ID,CV_Desc)
   PK : CV_ID

4. **StaffConsult**(Consult_ID,Consult_Name,Consult_username,Consult_Password, Consult_Email,Address,Qualification,Availability)
   PK : Consult_ID , Consult_Email

5. **StaffAdmin**(Staff_ID,Staff_Name,Staff_Password,Staff_Email,Staff_Address,Positio n)
   PK : Staff_ID

6. **StaffProof**(Staff_ID,Staff_Name,Staff_Password,Staff_Email,Staff_Address)
   PK : Staff_ID

7. **CompanyCV**(CV_ID ,Comp_ID)
   PK : CV_ID , Comp_ID
   FK : CV_ID reference CV(CV_ID)
   FK : Comp_ID reference Company(Comp_ID)

8. **Q&A**(Question_ID,Question_Description,Answer_ID,Answer_Description)
   PK : Question_ID , Answer_ID

9. **QuestionAdmin**(Question_ID ,Staff_ID)
   PK : Question_ID,Staff_ID
   FK : Staff_ID reference StaffAdmin(Staff_ID)
   FK : Question_ID reference Q&A(Question_ID)

10. **CustQ&A**(Question_ID ,Cust_ID)
    PK : Question_ID,Cust_ID
    FK : Cust_ID reference Customer(Cust_ID)
    FK : Question_ID reference Q&A(Question_ID)

11. **Staff_Contact**(Staff_Contact_Num,Staff_ID)
    PK : Staff_Contact_Num
    FK : Staff_ID reference
StaffConsult(Staff_ID),StaffAdmin(Staff_ID),StaffProof(Staff_ID)

12. **JobVacancy**(Job_Vacancy,Comp_ID)
    PK : Job_Vacancy
    FK : Comp_ID reference Company(Comp_ID)

13. **CustSkills**(Cust_Skills,Cust_ID)
    PK : Cust_Skills
    FK : Cust_ID reference Company(Cust_ID)

14. **Comp_Contact**(Comp_Contact_Num, Comp_ID)
    PK: Comp_Contact_Num
    FK: Comp_ID reference Company(Comp_ID)

15. **Cust_Contact**(Cust_Contact_Num, Cust_ID)
    PK: Cust_Contact_Num
    FK: Cust_ID reference Customer(Cust_ID)

16. **Packages**(Pack_ID, Pack_Name, Pack_Desc,Pack_Price)
    PK:Pack_ID

17. **Cust_Package**(Pack_ID, Cust_ID)
    PK: Pack_ID, Cust_ID
    FK: Pack_ID reference Packages (Pack_ID)
    FK: Cust_ID reference Customer (Cust_ID)

6.2 Normalization

1)Customer relation

1st NF

Customer (Cust_ID,Cust_Name,Cust_username,Cust_password,Cust_Email,Education,
Comp_ID,ConsultID)
PK : Cust_ID,Cust_Email
FK : Comp_ID reference COID(Comp_ID)
ConsultID reference StaffConsult(ConsultID)

2nd NF

CEmail(Cust_Email,Cust_username,Cust_password)
PK : Cust_Email

CID(Cust_ID,Cust_Name,Cust_Email,Education,Comp_ID,ConsultID)
PK : Cust_ID
FK : Cust_Email reference CEmail(Cust_Email)
Comp_ID reference COID(Comp_ID)
ConsultID reference StaffConsult(ConsultID)

Customer (Cust_ID,Cust_Email)
PK : Cust_ID,Cust_Email
FK : Cust_ID references CID(Cust_ID)
Cust_Email references CEmail(Cust_Email)

3rd NF

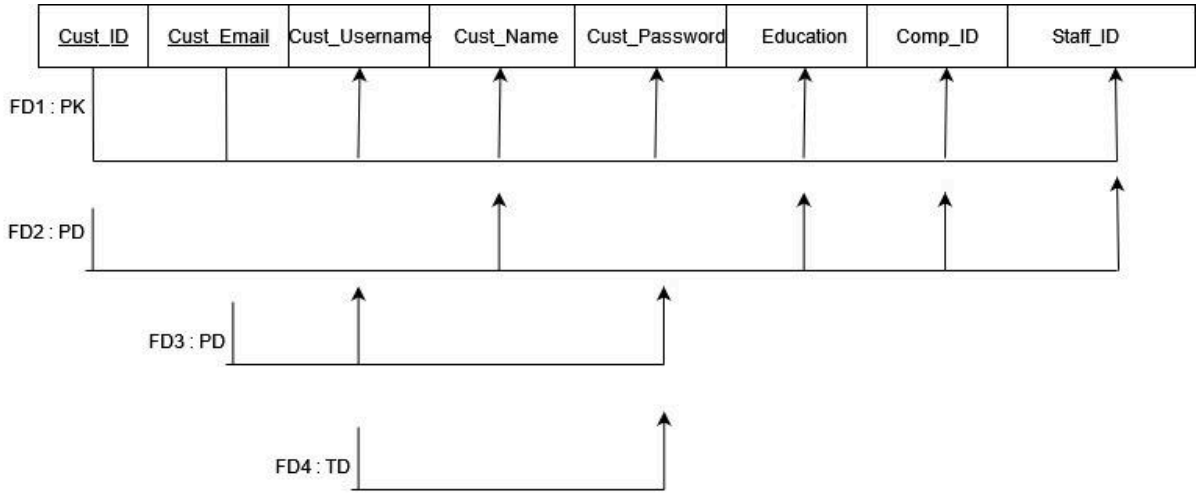
CUser(Cust_username,Cust_password)
PK: Cust_username

CEmail(Cust_Email,Cust_Username)
PK : Cust_Email
FK : Cust_username references CUser(Cust_username).

CID(Cust_ID,Cust_Name,Cust_Email,Education,Comp_ID,ConsultID)
PK : Cust_ID
FK : Cust_Email references CEmail(Cust_Email).
Comp_ID reference COID(Comp_ID)
ConsultID reference StaffConsult(ConsultID)

Customer (Cust_ID,Cust_Email)
PK : Cust_ID,Cust_Email
FK : Cust_ID references CID(Cust_ID)

Cust_Email references CEmail(Cust_Email)



2)Company relation

1st NF
Company(Comp_ID,Comp_Name,Comp_username,Comp_password,Comp_Email,Address)
PK: Comp_ID, Comp_Email

2nd NF
CoID(Comp_ID,Comp_Name,Comp_Email,Address)
PK: Comp_ID
FK: Comp_Email reference CoEmail(Comp_Email)

CoEmail(Comp_Email,Comp_username,Comp_password)
PK: Comp_Email

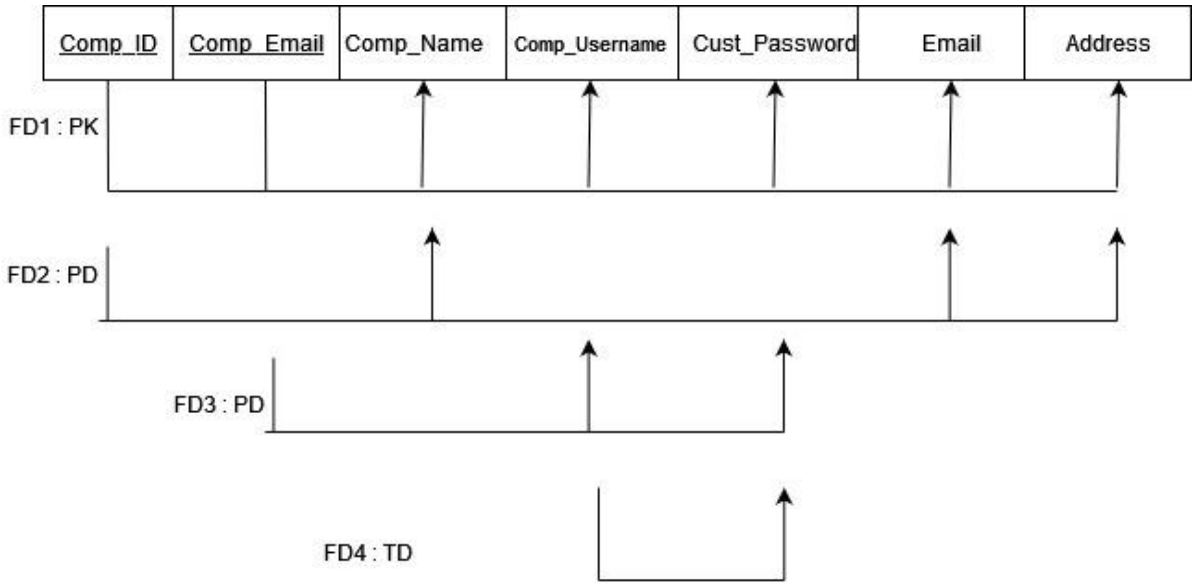
Company(Comp_ID,Comp_Email)
PK: Comp_ID, Comp_Email
FK: Comp_ID references CoID(Comp_ID)
Comp_Email references CoEmail(Comp_Email)

3rd NF
CoUser(Comp_username,Comp_password)
PK: Comp_username

CoID(Comp_ID,Comp_Name,Comp_Email,Address)
PK: Comp_ID

CoEmail(Comp_Email,Comp_username)
PK: Comp_Email
FK: Comp_username references CoUser(Comp_username)

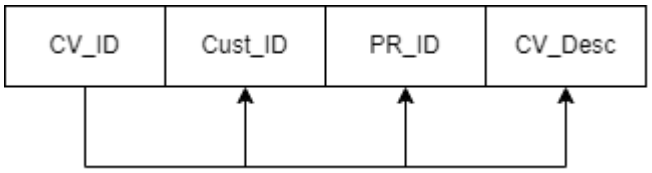
Company(Comp_ID,Comp_Email)
PK: Comp_ID, Comp_Email)
FK: Comp_ID references CoID(Comp_ID)
Comp_Email references CoEmail(Comp_Email)



3)CV relation

1st NF

CV(CV_ID , Cust_ID , PR_ID , CV_Desc)
PK: CV_ID



4)StaffConsult relation

1st NF

StaffConsult(Consult_ID , Consult_Name , Consult_username , Consult_Password , Consult_Email , Address , Qualification , Availability)
PK:Consult_ID , Consult_Email

2nd NF

ConsultantID(Consult_ID , Consult_Name , Address , Qualification , Availability)
PK:Consult_ID

ConsultantEmail(Consult_Email , Consult_username , Consult_Password)
PK:Consult_Email

StaffConsult(Consult_ID , Consult_Email)
PK:Consult_ID , Consult_Email)
FK:Consult_ID reference ConsultantID(Consult_ID)
FK:Consult_Email reference ConsultantEmail(Consult_Email)

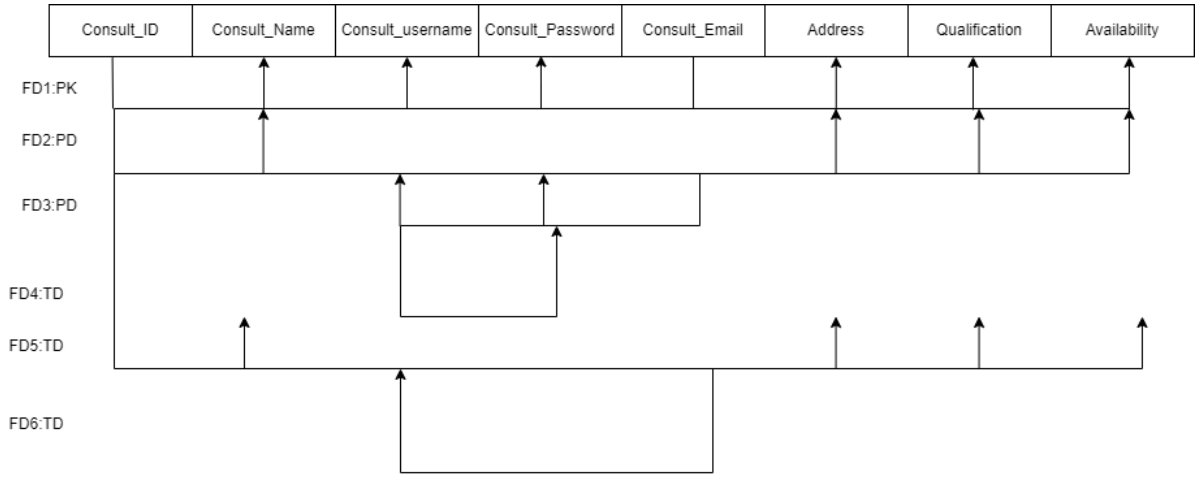
3rd NF

ConsultantUser(Consult_username , Consult_Password)
PK:Consult_username

ConsultantID(Consult_ID , Consult_Name , Address , Qualification , Availability)
PK:Consult_ID

ConsultantEmail(Consult_Email , Consult_username)
PK:Consult_Email
FK:Consult_username reference ConsultantUser(Consult_username)

StaffConsult(Consult_ID , Consult_Email)
PK:Consult_ID , Consult_Email)
FK:Consult_ID reference ConsultantID(Consult_ID)
FK:Consult_Email reference ConsultantEmail(Consult_Email)



5) StaffAdmin relation

1NF
StaffAdmin(Staff_ID,Staff_Email,Staff_Username,Staff_Name,Staff_Password,Staff_Address,Position)
PK : Staff_ID,Staff_Email

2NF
AdminID(Staff_ID,Staff_Name,Staff_Address,Position)
PK: Staff_ID
FK: Staff_Email reference AdminEmail(Staff_Email)

AdminEmail(Staff_Email,Staff_Username,Staff_Password,)
PK: Staff_Email

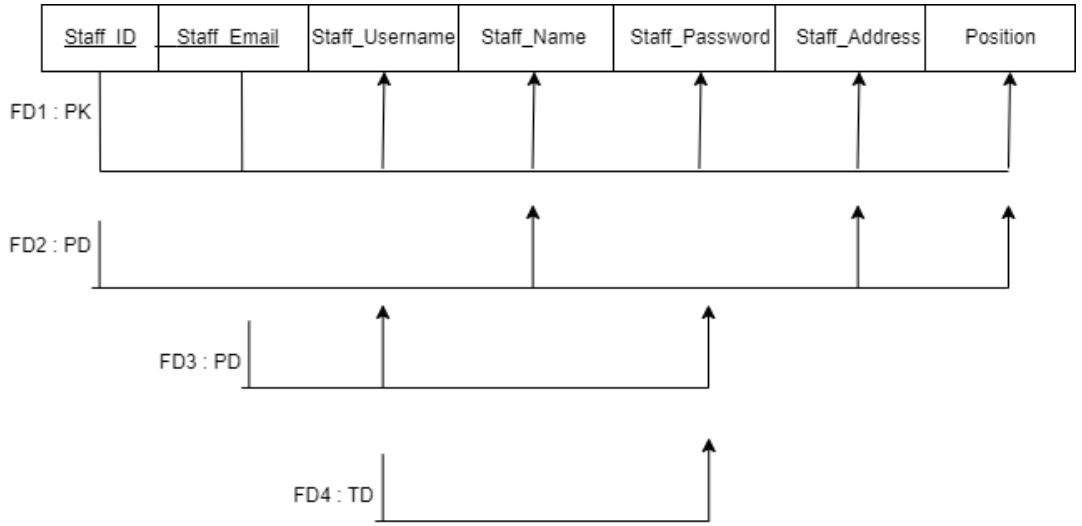
StaffAdmin(Staff_ID,Staff_Email)
PK: Staff_ID,Staff_Email
FK: Staff_ID reference AdminID (Staff_ID)
FK: Staff_Email references AdminEmail (Staff_Email)

3NF
AdminUser(Staff_Username, Staff_Password)
PK: Staff_Username

AdminID(Staff_ID,Staff_Name,Staff_Address,Position)
PK: Staff_ID

AdminEmail (Staff_Email , Staff_Username)
PK:Staff_Email
FK:Staff_Username reference AdminUser(Staff_Username)

StaffAdmin(Staff_ID , Staff_Email)
PK:Staff_ID , Staff_Email
FK:Staff_ID reference AdminID(Staff_ID)
FK:Staff_Email reference AdminEmail(Staff_Email)



6) StaffProof relation

1NF
StaffProof
(Staff_ID,Staff_Email,Staff_Username,Staff_Name,Staff_Password,Staff_Address)
PK : Staff_ID,Staff_Email

2NF
ProofID(Staff_ID,Staff_Name,Staff_Address)
PK: Staff_ID
FK: Staff_Email reference ProofEmail(Staff_Email)

ProofEmail(Staff_Email,Staff_Username,Staff_Password,)

PK: Staff_Email

StaffProof(Staff_ID,Staff_Email)
PK: Staff_ID,Staff_Email
FK: Staff_ID reference ProofID (Staff_ID)
FK: Staff_Email references ProofEmail (Staff_Email)

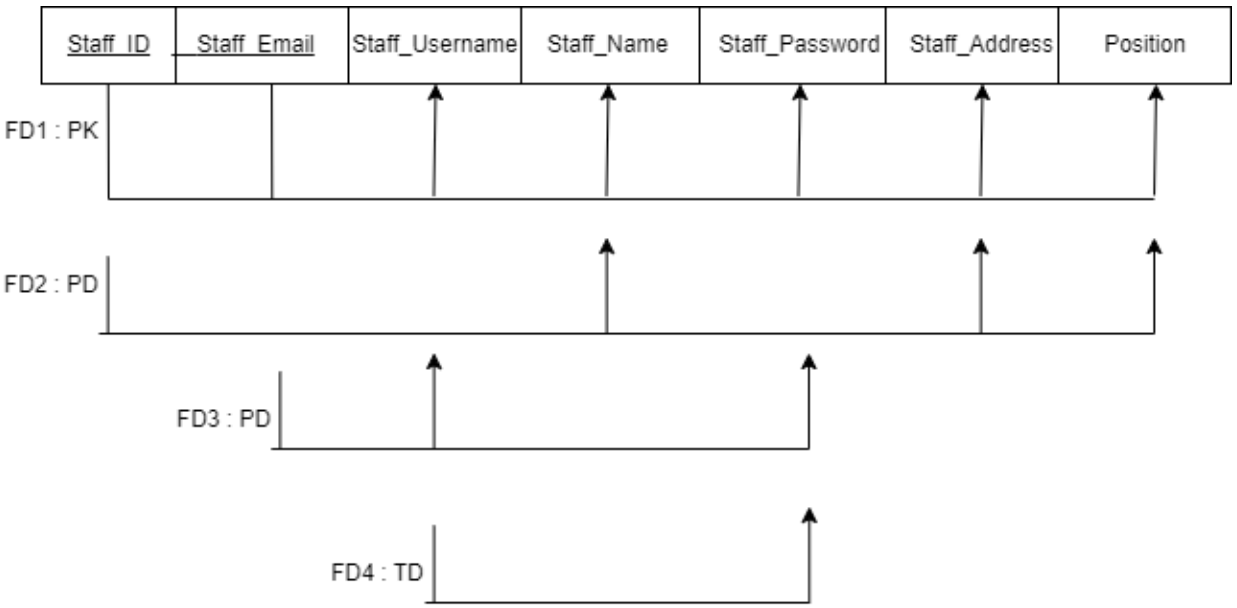
3NF

ProofUser(Staff_Username, Staff_Password)
PK: Staff_Username

ProofID(Staff_ID,Staff_Name,Staff_Address)
PK: Staff_ID
FK: Staff_Email reference ProofEmail(Staff_Email)

ProofEmail (Staff_Email , Staff_Username)
PK:Staff_Email
FK:Staff_Username reference ProofUser(Staff_Username)

StaffProof(Staff_ID , Staff_Email)
PK:Staff_ID , Staff_Email
FK:Staff_ID reference ProffID(Staff_ID)
FK:Staff_Email reference ProofEmail(Staff_Email)



7) CompanyCV

1NF

CompanyCV(CV_ID ,Comp_ID)

PK : Staff_ID

FK : Comp_ID reference COID(Comp_ID)

8) Q&A

1 NF

Q&A(Question_ID ,Question_Description , Answer_ID , Answer_Description)

Primary Key : Question_ID , Answer_ID

2 NF

Question(Question_ID ,Question_Description)

Primary Key : Question_ID

Answer(Answer_ID , Answer_Description)

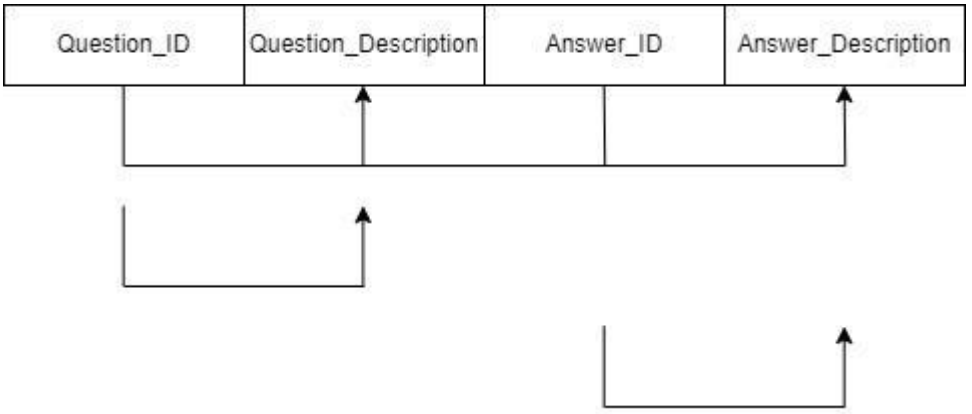
Primary Key : Answer_ID

Q&A(Question_ID , Answer_ID)

Primary Key : Question_ID , Answer_ID

Foreign Key : Question_ID reference Question(Question_ID)

: Answer_ID reference Answer(Answer_ID)



9) QuestionAdmin

1NF

QuestionAdmin(Question_ID ,Staff_ID)

PK : Question_ID

FK : Staff_ID reference Admin(Staff_ID)

10)CustQ&A

1NF

CustQ&A(Question_ID ,Cust_ID)

PK : Question_ID, Cust_ID

FK : Cust_ID reference CID(Cust_ID)

FK: Question_ID reference Question(Question_ID)

11) AdminContact

1NF

AdminContact(Contact,Staff_ID)

PK : Contacta

FK : Staff_ID reference StaffConsult(Staff_ID),StaffAdmin(Staff_ID),StaffProof(Staff_ID)

12) ConsultContact

1NF

ConsultContact(Contact,Staff_ID)

PK : Contact

FK : Staff_ID reference StaffConsult(Staff_ID),StaffAdmin(Staff_ID),StaffProof(Staff_ID)

13) ProofContact

1NF

ProofContact(Contact,Staff_ID)

PK : Contact

FK : Staff_ID reference StaffConsult(Staff_ID),StaffAdmin(Staff_ID),StaffProof(Staff_ID)

14)JobVacancy

1NF

JobVacancy(Job_Vacancy,Comp_ID)

PK : Job_Vacancy

FK : Comp_ID reference COID(Comp_ID)

15)CustSkills

1NF
CustSkills(Cust_Skills,Cust_ID)
PK : Cust_Skills
FK : Cust_ID reference CID(Cust_ID)

16)Comp_Contact

1NF
Comp_Contact(Comp_Contact_Num, Comp_ID)
PK: Comp_Contact_Num
FK: Comp_ID reference COID(Comp_ID)

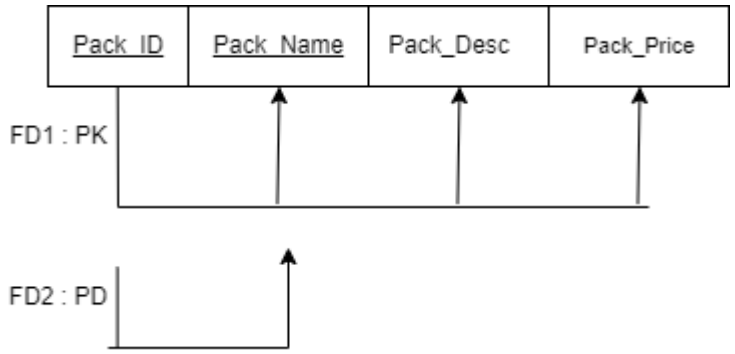
17)Cust_Contact

1NF
Cust_Contact(Cust_Contact_Num, Cust_ID)
PK: Cust_Contact_Num
FK: Cust_ID reference CID(Cust_ID)

18) **Package**

1NF
Package(Pack_ID, Pack_Name, Pack_Desc,Pack_Price)
PK:Pack_ID

2NF
PackID (Pack_ID, Pack_Name)
PK:Pack_ID
FK: Pack_ID reference Package(Pack_ID)



7.0 SQL Commands

COMPANY

CoUser

```
--create table for company username to store username and password
create table CoUser(
Comp_username varchar2(50),
Comp_password varchar2 (50),
CONSTRAINT comp_username_pk PRIMARY KEY (Comp_username)
);

--insert values into table CoUser
INSERT INTO CoUser VALUES ('Tech Innovators Inc', 'pass123');
INSERT INTO CoUser VALUES ('Data Masters Ltd', 'secureTech456');
INSERT INTO CoUser VALUES ('Cloud Tech Solutions', 'alphaPass789');
INSERT INTO CoUser VALUES ('Innova Systems LLC', 'gtco2022');
INSERT INTO CoUser VALUES ('Rapid Systems Co', 'webPass123');
```

CoEmail

```
--create table for company email to store username and email
create table CoEmail(
Comp_Email varchar2(50),
Comp_username varchar2(50),
CONSTRAINT Comp_Email_pk PRIMARY KEY(Comp_Email),
CONSTRAINT Comp_Email_fk FOREIGN KEY (Comp_username) REFERENCES
CoUser(Comp_username)
```

```
);
--insert values into table CoEmail
INSERT INTO CoEmail VALUES ('techinnovators@example.com', 'Tech Innovators Inc');
INSERT INTO CoEmail VALUES ('datamasters@example.com', 'Data Masters Ltd');
INSERT INTO CoEmail VALUES ('cloudtech@example.com', 'Cloud Tech Solutions');
INSERT INTO CoEmail VALUES ('innova@example.com', 'Innova Systems LLC');
INSERT INTO CoEmail VALUES ('rapidsystems@example.com', 'Rapid Systems Co');
```

CoID

```
create table CoID(
Comp_ID varchar2 (5),
Comp_Name varchar2 (50),
Comp_Email varchar2 (50),
Address varchar2 (150),
CONSTRAINT Comp_ID_pk PRIMARY KEY(Comp_ID),
CONSTRAINT comid_fk FOREIGN KEY(Comp_Email) REFERENCES
CoEmail(Comp_Email)
);

INSERT INTO CoID VALUES ('C0001', 'Tech Innovators Inc',
'techinnovators@example.com', '123 Tech Street, Tech City');
INSERT INTO CoID VALUES ('C0002', 'Data Masters Ltd', 'datamasters@example.com',
'456 Data Avenue, Data City');
INSERT INTO CoID VALUES ('C0003', 'Cloud Tech Solutions', 'cloudtech@example.com',
'789 Cloud Lane, Cloud City');
INSERT INTO CoID VALUES ('C0004', 'Innova Systems LLC', 'innova@example.com',
'101 Innovate Road, Innovate City');
INSERT INTO CoID VALUES ('C0005', 'Rapid Systems Co', 'rapidsystems@example.com',
'202 Rapid Street, Rapid City');
```

Company

```
-- Company
--Table to store Company Username & Password
create table Company(
Comp_ID varchar2 (5),
Comp_Email varchar2 (50),
CONSTRAINT Company_pk PRIMARY KEY ( Comp_ID,Comp_Email ),
CONSTRAINT compId_fk FOREIGN KEY(Comp_ID) REFERENCES CoID(Comp_ID),
CONSTRAINT compemail_fk FOREIGN KEY(Comp_Email) REFERENCES
CoEmail(Comp_Email)
);
--Insert values into table CoUser
INSERT INTO Company VALUES ('C0001', 'techinnovators@example.com');
INSERT INTO Company VALUES ('C0002', 'datamasters@example.com');
INSERT INTO Company VALUES ('C0003', 'cloudtech@example.com');
INSERT INTO Company VALUES ('C0004', 'innova@example.com');
INSERT INTO Company VALUES ('C0005', 'rapidsystems@example.com');
```

JobVacancy

```
--Table to store Company ID & Job offer by the company
Create table JobVacancy(
   Job_Vacancy varchar(50),
   Comp_ID varchar(5),
   CONSTRAINT job_pk PRIMARY KEY (Job_Vacancy),
   CONSTRAINT job_fk FOREIGN KEY (Comp_ID) REFERENCES CoID (Comp_ID)
);
--Insert values into table JobVacancy
INSERT INTO JobVacancy(Job_Vacancy,Comp_ID)
VALUES ('Software Developer', 'C0001');
```

```
INSERT INTO JobVacancy(Job_Vacancy,Comp_ID)
VALUES ('Marketing Specialist', 'C0002');
INSERT INTO JobVacancy(Job_Vacancy,Comp_ID)
VALUES ('Financial Analyst', 'C0003');
INSERT INTO JobVacancy(Job_Vacancy,Comp_ID)
VALUES ('Customer Service Representative', 'C0004');
INSERT INTO JobVacancy(Job_Vacancy,Comp_ID)
VALUES ('Graphic Designer', 'C0005');
```

Comp_Contact

```
--Table to store Company phone number & ID
Create table Comp_Contact(
   Comp_Contact_Num varchar(50),
   Comp_ID varchar(5),
   CONSTRAINT compcontact_pk PRIMARY KEY (Comp_Contact_Num),
   CONSTRAINT compcontact_fk FOREIGN KEY (Comp_ID) REFERENCES CoID
(Comp_ID)
);
--Insert values into table Comp_Contact
INSERT INTO Comp_Contact(Comp_Contact_Num, Comp_ID)
VALUES ('012 456 7890','C0001');
INSERT INTO Comp_Contact(Comp_Contact_Num, Comp_ID)
VALUES ('011 233 4455','C0002');
INSERT INTO Comp_Contact(Comp_Contact_Num, Comp_ID)
VALUES ('017 876 5432','C0003');
INSERT INTO Comp_Contact(Comp_Contact_Num, Comp_ID)
VALUES ('014 578 9012','C0004');
INSERT INTO Comp_Contact(Comp_Contact_Num, Comp_ID)
VALUES ('013 875 4321','C0005');
```

CV

```
-- CV

--Table to store CV ID , Customer ID , Proofreader ID , CV Description

CREATE TABLE CV (
	CV_ID VARCHAR(20),
	Cust_ID VARCHAR(20),
	PR_ID VARCHAR(5),
	CV_Desc VARCHAR2(150),
	CONSTRAINT cv_pk PRIMARY KEY (CV_ID),
	CONSTRAINT cv_fk FOREIGN KEY (Cust_ID) REFERENCES CID(Cust_ID),
	CONSTRAINT cv2_fk FOREIGN KEY (PR_ID) REFERENCES ProofID(Staff_ID)
);

--Add Admin ID attribute to the CV table
ALTER TABLE CV
ADD admID VARCHAR2(5);
ALTER TABLE CV
ADD CONSTRAINT cvadm_fk FOREIGN KEY (admID) REFERENCES
AdminID(Staff_ID);

UPDATE CV
SET admID = 'X0001'
WHERE CV_ID = 'K1';

UPDATE CV
SET admID = 'X0001'
WHERE CV_ID = 'K2';

UPDATE CV
SET admID = 'X0002'
WHERE CV_ID = 'K3';

UPDATE CV
SET admID = 'X0003'
WHERE CV_ID = 'K4';

UPDATE CV
SET admID = 'X0004'
WHERE CV_ID = 'K5';

INSERT INTO CV VALUES ('K1', 'Z0001', 'P0021', 'Experienced manager in the finance
sector');
```

```
INSERT INTO CV VALUES ('K2', 'Z0002', 'P0022', 'Detail-oriented assistant with strong organizational skills');
INSERT INTO CV VALUES ('K3', 'Z0003', 'P0023', 'Clerical professional with excellent communication skills');
INSERT INTO CV VALUES ('K4', 'Z0004', 'P0024', 'Supervisor with a track record of team leadership');
INSERT INTO CV VALUES ('K5', 'Z0005', 'P0025', 'Analytical mindset for data-driven decision-making');
```

CV_SK

```
--Table to store CV ID & CV_Skills
CREATE TABLE CV_SK
(
CV_ID VARCHAR2(5),
CV_Skills VARCHAR2(150),
CONSTRAINT sk_pk PRIMARY KEY (CV_Skills,CV_ID),
CONSTRAINT sk_fk FOREIGN KEY (CV_ID) REFERENCES CV(CV_ID)
);


--Insert values into table CV
INSERT INTO CV_SK VALUES ('K1','Excel Expert');
INSERT INTO CV_SK VALUES ('K1','Python Data Analysis');
INSERT INTO CV_SK VALUES ('K2','AI Search Engine Expert');
INSERT INTO CV_SK VALUES ('K3','C++ Expert');
INSERT INTO CV_SK VALUES ('K3','Video Capturing');
```

Consultant
ConsutltantUser

```
--Consultant
--Table to store Consultant Username & Password
CREATE TABLE ConsultantUser (
	Consult_username VARCHAR(50),
	Consult_Password VARCHAR(20),
	CONSTRAINT consultantuser_pk PRIMARY KEY (Consult_username)
);
```

```
--Insert values into table ConsultantUser
INSERT INTO ConsultantUser (Consult_username, Consult_Password) VALUES
('johnsmith123', 'smithpass');
INSERT INTO ConsultantUser (Consult_username, Consult_Password) VALUES
('maryjones456', 'jonespass');
INSERT INTO ConsultantUser (Consult_username, Consult_Password) VALUES
('robertbrown789', 'brownpass');
INSERT INTO ConsultantUser (Consult_username, Consult_Password) VALUES
('emilywhite234', 'whitepass');
INSERT INTO ConsultantUser (Consult_username, Consult_Password) VALUES
('davidlee567', 'leepass');
```

ConsultantID

```
--Table to store Consultant ID , Name , Address , Qualification & Availability
CREATE TABLE ConsultantID (
	Consult_ID VARCHAR(20),
	Consult_Name VARCHAR(50),
	Address VARCHAR(50),
	Qualification VARCHAR(50),
	Availability DATE,
	CONSTRAINT consultantid_pk PRIMARY KEY (Consult_ID)
);

--Insert values into table ConsultantID
INSERT INTO ConsultantID (Consult_ID, Consult_Name, Address, Qualification,
Availability)
VALUES ('K1', 'John Smith', '123 Main St', 'MBA', TO_DATE('2023-05-15',
'YYYY-MM-DD'));
INSERT INTO ConsultantID (Consult_ID, Consult_Name, Address, Qualification,
Availability)
VALUES ('K2', 'Mary Jones', '456 Oak Ave', 'PhD', TO_DATE('2023-08-22',
'YYYY-MM-DD'));
INSERT INTO ConsultantID (Consult_ID, Consult_Name, Address, Qualification,
Availability)
VALUES ('K3', 'Robert Brown', '789 Maple Ln', 'BSc', TO_DATE('2023-04-10',
'YYYY-MM-DD'));

INSERT INTO ConsultantID (Consult_ID, Consult_Name, Address, Qualification,
Availability)
VALUES ('K4', 'Emily White', '234 Pine Rd', 'MS', TO_DATE('2023-11-30',
'YYYY-MM-DD'));
INSERT INTO ConsultantID (Consult_ID, Consult_Name, Address, Qualification,
Availability)
```

```
VALUES ('K5', 'David Lee', '567 Elm Blvd', 'MBA', TO_DATE('2023-06-18',
'YYYY-MM-DD'));
```

ConsultantEmail

```
--Table to store Consultant Email & Username
CREATE TABLE ConsultantEmail (
	Consult_Email VARCHAR(50),
	Consult_username VARCHAR(50),
	CONSTRAINT consultantemail_pk PRIMARY KEY (Consult_Email),
	CONSTRAINT consultantemail_fk FOREIGN KEY (Consult_username)
REFERENCES ConsultantUser (Consult_username)
);

--Insert values into table ConsultantEmail
INSERT INTO ConsultantEmail (Consult_Email, Consult_username) VALUES
('john.smith@example.com', 'johnsmith123');
INSERT INTO ConsultantEmail (Consult_Email, Consult_username) VALUES
('mary.jones@example.com', 'maryjones456');
INSERT INTO ConsultantEmail (Consult_Email, Consult_username) VALUES
('robert.brown@example.com', 'robertbrown789');
INSERT INTO ConsultantEmail (Consult_Email, Consult_username) VALUES
('emily.white@example.com', 'emilywhite234');
INSERT INTO ConsultantEmail (Consult_Email, Consult_username) VALUES
('david.lee@example.com', 'davidlee567');
```

StaffConsult

```
--Table to store Consultant ID & Email
CREATE TABLE StaffConsult (
	Consult_ID VARCHAR(20),
	Consult_Email VARCHAR(50),
	CONSTRAINT staffconsult_pk PRIMARY KEY (Consult_ID, Consult_Email),
	CONSTRAINT staffconsult2_fk FOREIGN KEY (Consult_ID) REFERENCES
ConsultantID(Consult_ID),
	CONSTRAINT staffconsult3_fk FOREIGN KEY (Consult_Email) REFERENCES
ConsultantEmail(Consult_Email)
);
```

```
--Insert values into table StaffConsultant
INSERT INTO StaffConsult (Consult_ID, Consult_Email)
VALUES ('K1', 'john.smith@example.com');
INSERT INTO StaffConsult (Consult_ID, Consult_Email)
VALUES ('K2', 'mary.jones@example.com');
INSERT INTO StaffConsult (Consult_ID, Consult_Email)
VALUES ('K3', 'robert.brown@example.com');
INSERT INTO StaffConsult (Consult_ID, Consult_Email)
VALUES ('K4', 'emily.white@example.com');
INSERT INTO StaffConsult (Consult_ID, Consult_Email)
VALUES ('K5', 'david.lee@example.com');
```

ConsultContact

```
--Create table consultant contact
CREATE TABLE ConsultContact
(
Staff_ID VARCHAR2(5),
Contact VARCHAR2(15),
CONSTRAINT cc_pk PRIMARY KEY (Contact),
CONSTRAINT cid_fk FOREIGN KEY (Staff_ID) REFERENCES
ConsultantID(Consult_ID)
);
INSERT INTO ConsultContact VALUES ('K1','011 0000000');
INSERT INTO ConsultContact VALUES ('K2','017 1111111');
INSERT INTO ConsultContact VALUES ('K3','018 2222222');
INSERT INTO ConsultContact VALUES ('K4','018 3333333');
INSERT INTO ConsultContact VALUES ('K5','019 4444444');
```

Admin

AdminUser

```
-- Admin
--Table to store Admin Username & Password
create table AdminUser(
        Staff_Username varchar (50),
        Staff_Password varchar (50),
        CONSTRAINT adminuser_pk PRIMARY KEY (Staff_UserName)
```

```
);

--Insert values into table AdminUser
INSERT INTO AdminUser (Staff_Username,Staff_Password)
Values ('johndoe123', 'pass123');
INSERT INTO AdminUser (Staff_Username,Staff_Password)
Values ('janesmith456', 'securepass');
INSERT INTO AdminUser (Staff_Username,Staff_Password)
Values ( 'mikejones789', 'p@ssw0rd');
INSERT INTO AdminUser (Staff_Username,Staff_Password)
Values ('sarahwhite321', 's3cur3password');
INSERT INTO AdminUser (Staff_Username,Staff_Password)
Values ('robertgreen555', 'greenapple');
```

AdminID

```
--Table to store Admin username & password
create table AdminID(
	Staff_ID varchar (5),
	Staff_Name varchar (50),
	Staff_Address varchar (50),
	Position varchar (50),
	CONSTRAINT adminid_pk PRIMARY KEY (Staff_ID)
);

--Insert values into table AdminID
INSERT INTO AdminID (Staff_ID,Staff_Name,Staff_Address,Position)
Values ('X0001', 'John Doe', '123 Main St', 'Manager');
INSERT INTO AdminID (Staff_ID,Staff_Name,Staff_Address,Position)
Values ('X0002', 'Jane Smith','456 Oak Ave', 'Assistant');
INSERT INTO AdminID (Staff_ID,Staff_Name,Staff_Address,Position)
Values ('X0003', 'Mike Jones', '789 Elm St', 'Clerk');
INSERT INTO AdminID (Staff_ID,Staff_Name,Staff_Address,Position)
Values ('X0004', 'Sarah White', '321 Pine Rd', 'Supervisor');
INSERT INTO AdminID (Staff_ID,Staff_Name,Staff_Address,Position)
Values ('X0005', 'Robert Green', '555 Cedar Ln', 'Analyst');
```

AdminEmail

```
--Table to store Admin Email & Username
create table AdminEmail(
	Staff_Email varchar (50),
	Staff_Username varchar (50),
	CONSTRAINT adminemail_pk PRIMARY KEY (Staff_Email),
	CONSTRAINT adminemail_fk FOREIGN KEY (Staff_Username) REFERENCES
AdminUser (Staff_Username)
);

--Insert values into table AdminEmail
INSERT INTO AdminEmail (Staff_Email,Staff_Username)
Values ('john.doe@example.com', 'johndoe123');
INSERT INTO AdminEmail (Staff_Email,Staff_Username)
Values ( 'jane.smith@example.com', 'janesmith456');
INSERT INTO AdminEmail (Staff_Email,Staff_Username)
Values ('mike.jones@example.com', 'mikejones789');
INSERT INTO AdminEmail (Staff_Email,Staff_Username)
Values ( 'sarah.white@example.com', 'sarahwhite321');
INSERT INTO AdminEmail (Staff_Email,Staff_Username)
Values ('robert.green@example.com', 'robertgreen555');
```

StaffAdmin

```
--Table to store Admin ID & Email
create table StaffAdmin(
	Staff_ID varchar (5),
	Staff_Email varchar (50),
	CONSTRAINT admin_pk PRIMARY KEY (Staff_ID , Staff_Email),
	CONSTRAINT admin1_fk FOREIGN KEY (Staff_ID) REFERENCES AdminID
(Staff_ID),
	CONSTRAINT admin2_fk FOREIGN KEY (Staff_Email) REFERENCES
AdminEmail (Staff_Email)
);

--Insert values into table StaffAdmin
INSERT INTO StaffAdmin (Staff_ID,Staff_Email)
Values ( 'X0001', 'john.doe@example.com');
INSERT INTO StaffAdmin (Staff_ID,Staff_Email)
Values ( 'X0002', 'jane.smith@example.com');
INSERT INTO StaffAdmin (Staff_ID,Staff_Email)
Values ('X0003', 'mike.jones@example.com');
```

```
INSERT INTO StaffAdmin (Staff_ID,Staff_Email)
Values ( 'X0004', 'sarah.white@example.com');
INSERT INTO StaffAdmin (Staff_ID,Staff_Email)
Values ( 'X0005', 'robert.green@example.com');
```

AdminContact

```
--create table of admin contact
CREATE TABLE AdminContact
(
Staff_ID VARCHAR2(5),
Contact VARCHAR2(15),
CONSTRAINT ac_pk PRIMARY KEY (Contact),
CONSTRAINT aid_fk FOREIGN KEY (Staff_ID) REFERENCES AdminID (Staff_ID)
);

INSERT INTO AdminContact VALUES ('X0001','017 0000231');
INSERT INTO AdminContact VALUES ('X0002','012 2341231');
INSERT INTO AdminContact VALUES ('X0003','015 0912312');
INSERT INTO AdminContact VALUES ('X0004','016 1233212');
INSERT INTO AdminContact VALUES ('X0005','018 0988909');
```

ProofReader

ProofUser

```
-- Proofreader
--Table to store Proofreader username & password
create table ProofUser(
        Staff_Username varchar (50),
        Staff_Password varchar (50),
        CONSTRAINT proofuser_pk PRIMARY KEY (Staff_UserName)
);


--Insert values into table ProofUser
INSERT INTO ProofUser (Staff_Username,Staff_Password)
Values ('alexsmith123', 'password123');
INSERT INTO ProofUser (Staff_Username,Staff_Password)
```

```
Values ('sophiejones456', 'securepassword');
INSERT INTO ProofUser (Staff_Username,Staff_Password)
Values ('michaelbrown789', 'brownpass');
INSERT INTO ProofUser (Staff_Username,Staff_Password)
Values ('oliviawilson321', 'wilsonpass');
INSERT INTO ProofUser (Staff_Username,Staff_Password)
Values ('ryantaylor555', 'taylor123');
```

<u>ProofID</u>

```
--Table to store Proofreader ID , Name , Address
create table ProofID(
        Staff_ID varchar (5),
        Staff_Name varchar (50),
        Staff_Address varchar (50),
        Position varchar (50),
        CONSTRAINT proofid_pk PRIMARY KEY (Staff_ID)
);

UPDATE ProofID
SET Adm_ID = 'X0001'
WHERE Staff_ID = 'P0021';
UPDATE ProofID
SET Adm_ID = 'X0002'
WHERE Staff_ID = 'P0022';
UPDATE ProofID
SET Adm_ID = 'X0002'
WHERE Staff_ID = 'P0023';
UPDATE ProofID
SET Adm_ID = 'X0003'
WHERE Staff_ID = 'P0024';
UPDATE ProofID
SET Adm_ID = 'X0004'
WHERE Staff_ID = 'P0025';

--Add another attribute (Admin ID) into ProofID table
ALTER TABLE ProofID
ADD Adm_ID VARCHAR2(5);


--Insert values into table ProofID
INSERT INTO ProofID (Staff_ID,Staff_Name,Staff_Address,Position)
Values ('P0021', 'Alex Smith', '123 Oak St', 'Manager');
```

```
INSERT INTO ProofID (Staff_ID,Staff_Name,Staff_Address,Position)
Values ('P0022', 'Sophie Jones', '456 Elm Ave', 'Assistant');

INSERT INTO ProofID (Staff_ID,Staff_Name,Staff_Address,Position)
Values ('P0023', 'Michael Brown', '789 Birch Ln', 'Clerk');

INSERT INTO ProofID (Staff_ID,Staff_Name,Staff_Address,Position)
Values ('P0024', 'Olivia Wilson', '321 Cedar Rd', 'Supervisor');

INSERT INTO ProofID (Staff_ID,Staff_Name,Staff_Address,Position)
Values ('P0025', 'Ryan Taylor', '555 Maple Dr', 'Analyst');
```

<u>ProofEmail</u>

```
--Table to store Proofreader Email & Password
create table ProofEmail(
	Staff_Email varchar (50),
	Staff_Username varchar (50),
	CONSTRAINT proofemail_pk PRIMARY KEY (Staff_Email),
	CONSTRAINT proofemail_fk FOREIGN KEY (Staff_Username) REFERENCES
ProofUser (Staff_Username)
);

--Insert values into table ProofEmail
INSERT INTO ProofEmail (Staff_Email,Staff_Username)
Values ('alex.smith@example.com', 'alexsmith123');
INSERT INTO ProofEmail (Staff_Email,Staff_Username)
Values ('sophie.jones@example.com', 'sophiejones456');
INSERT INTO ProofEmail (Staff_Email,Staff_Username)
Values ('michael.brown@example.com', 'michaelbrown789');
INSERT INTO ProofEmail (Staff_Email,Staff_Username)
Values ('olivia.wilson@example.com', 'oliviawilson321');
INSERT INTO ProofEmail (Staff_Email,Staff_Username)
Values ('ryan.taylor@example.com', 'ryantaylor555');
```

StaffProof

```
--Table to store Staff ID & Email
create table StaffProof(
	Staff_ID varchar (5),
	Staff_Email varchar (50),

	CONSTRAINT proof_pk PRIMARY KEY (Staff_ID,Staff_Email),
	CONSTRAINT proof1_fk FOREIGN KEY (Staff_ID) REFERENCES ProofID
(Staff_ID),
	CONSTRAINT proof2_fk FOREIGN KEY (Staff_Email) REFERENCES ProofEmail
(Staff_Email)

);

--Insert values into table StaffProof
INSERT INTO StaffProof (Staff_ID,Staff_Email)
Values ('P0021', 'alex.smith@example.com');
INSERT INTO StaffProof (Staff_ID,Staff_Email)
Values ('P0022', 'sophie.jones@example.com');
INSERT INTO StaffProof (Staff_ID,Staff_Email)
Values ('P0023', 'michael.brown@example.com');
INSERT INTO StaffProof (Staff_ID,Staff_Email)
Values ('P0024', 'olivia.wilson@example.com');
INSERT INTO StaffProof (Staff_ID,Staff_Email)
Values ('P0025', 'ryan.taylor@example.com');


--create table for proof contact
CREATE TABLE ProofContact
(
Staff_ID VARCHAR2(5),
Contact VARCHAR2(15),
CONSTRAINT prof_pk PRIMARY KEY (Contact),
CONSTRAINT prof_ct_fk FOREIGN KEY (Staff_ID) REFERENCES ProofID(Staff_ID)
);

INSERT INTO ProofContact VALUES ('P0021','012 2342143');
INSERT INTO ProofContact VALUES ('P0022','013 3456654');
INSERT INTO ProofContact VALUES ('P0023','014 2222222');
INSERT INTO ProofContact VALUES ('P0024','018 3333333');
INSERT INTO ProofContact VALUES ('P0025','019 0987789');
```

QNA

<u>Question</u>

```
--Table to store Question ID & Description
CREATE TABLE Question(
	question_id VARCHAR2(5) CONSTRAINT quest_q_id_pk PRIMARY KEY,
	question_description VARCHAR2(100)
);

--Insert values into table Question
Insert INTO Question
VALUES ('Q0001','What are the payment methods available');
Insert INTO Question VALUES
('Q0002','How can I customize my CV');
Insert INTO Question VALUES
('Q0003','Can I import my LinkedIn profile');
Insert INTO Question VALUES
('Q0004','How do I export my CV');
Insert INTO Question VALUES
('Q0005','Can I track how many people viewed my CV');
```

<u>Answer</u>

```
--Table to store Answer ID , Description
CREATE TABLE Answer(
	answer_id VARCHAR2(5) CONSTRAINT ans_a_id_pk PRIMARY KEY,
	answer_description VARCHAR2(100)
);


INSERT INTO Answer VALUES ('A0001' , 'There are two payment methods available');
INSERT INTO Answer VALUES ('A0002', 'We offer a variety of CV templates, including
professional, modern, and creative styles');
INSERT INTO Answer VALUES ('A0003', 'You can customize your CV');
INSERT INTO Answer VALUES ('A0004', 'You can import your LinkedIn profile to quickly
create a CV');
INSERT INTO Answer VALUES ('A0005', 'You can export your CV in PDF format');
```

QA

```
--Create table QA with question ID and Answer ID
CREATE TABLE QA(
	quest_id VARCHAR2(5),
	ans_id VARCHAR2(5),
	CONSTRAINT qa_pk PRIMARY KEY (quest_id , ans_id),
	CONSTRAINT qa_qid_fk FOREIGN KEY(quest_id) REFERENCES
Question(question_id),
	CONSTRAINT qa_aid_fk FOREIGN KEY(ans_id) REFERENCES
Answer(answer_id)
);


--Insert values into table QA
INSERT INTO QA VALUES ('Q0001','A0001');
INSERT INTO QA VALUES ('Q0002','A0002');
INSERT INTO QA VALUES ('Q0003','A0003');
INSERT INTO QA VALUES ('Q0004','A0004');
INSERT INTO QA VALUES ('Q0005','A0005');
```

CustQA

```
--Table to store CustQA which  have Question ID , Customer ID
CREATE TABLE CustQA
(
	que_id VARCHAR2(5),
	cus_id VARCHAR2(5),
	ans_id VARCHAR2(5),
	CONSTRAINT cqa_pk PRIMARY KEY(que_id),
	CONSTRAINT q_fk FOREIGN KEY(que_id) REFERENCES
Question(question_id),
	CONSTRAINT cust_fk FOREIGN KEY (cus_id) REFERENCES CID (Cust_ID),
	CONSTRAINT cuans_fk FOREIGN KEY (ans_id) REFERENCES
Answer(answer_id)
);

--Insert value into CustQA table
INSERT INTO CustQA VALUES ('Q0001','Z0001','A0001');
INSERT INTO CustQA VALUES ('Q0002','Z0001','A0002');
INSERT INTO CustQA VALUES ('Q0003','Z0002','A0003');
INSERT INTO CustQA VALUES ('Q0004','Z0003','A0004');
INSERT INTO CustQA VALUES ('Q0005','Z0003','A0005');
```

AdminQA

```
--Table to store AdminQA which have Question ID , Answer ID
CREATE TABLE AdminQA
(
	que_id VARCHAR2(5),
	ans_id VARCHAR2(5),
	admin_id VARCHAR2(5),
	CONSTRAINT ad_qa_pk PRIMARY KEY (que_id),
	CONSTRAINT a_fk FOREIGN KEY (que_id) REFERENCES
Question(question_id),
	CONSTRAINT aa_id_fk FOREIGN KEY (admin_id) REFERENCES
AdminID(Staff_ID),
	CONSTRAINT aN_id_fk FOREIGN KEY (ans_id) REFERENCES
Answer(answer_id)
);

--Insert values into table AdminQA
INSERT INTO AdminQA VALUES ('Q0001','A0001','X0003');
INSERT INTO AdminQA VALUES ('Q0002','A0002','X0005');
INSERT INTO AdminQA VALUES ('Q0003','A0003','X0002');
INSERT INTO AdminQA VALUES ('Q0004','A0004','X0004');
INSERT INTO AdminQA VALUES ('Q0005','A0005','X0003');
```

Customer

CUser

```
--Table to store Customer Email , Username
create table CUser(
Cust_username varchar2(50),
Cust_password varchar2 (50),
CONSTRAINT cust_username_pk PRIMARY KEY (Cust_username)
);


--Insert values into table CEmail
INSERT INTO CUser VALUES ('user1', 'pass123');
INSERT INTO CUser VALUES ('john_doe', 'securePW456');
INSERT INTO CUser VALUES ('alice_smith', '12345pass');
INSERT INTO CUser VALUES ('jdoe2022', 'testpass789');
INSERT INTO CUser VALUES ('admin_user', 'adminPass123');
```

CEmail

```
--Table to store Customer ID , Name , Education , Company ID , Consultant ID
create table CEmail(
Cust_Email varchar2(50),
Cust_Username varchar2(50),
CONSTRAINT Cust_Email_pk PRIMARY KEY(Cust_Email),
CONSTRAINT Cust_Email_fk FOREIGN KEY (Cust_Username) REFERENCES
CUser(Cust_Username)
);

--Insert values into table CID
INSERT INTO CEmail VALUES ('user1@example.com', 'user1');
INSERT INTO CEmail VALUES ('john.doe@email.com', 'john_doe');
INSERT INTO CEmail VALUES ('alice.smith@email.com', 'alice_smith');
INSERT INTO CEmail VALUES ('jdoe2022@email.com', 'jdoe2022');
INSERT INTO CEmail VALUES ('admin.user@email.com', 'admin_user');
```

CID

```
--Create table for customer ID
create table CID(
Cust_ID varchar2 (5),
Cust_Name varchar2 (50),
Education varchar2 (50),
Comp_ID varchar2 (5),
StaffID varchar2 (5),
CONSTRAINT Cust_ID_pk PRIMARY KEY(Cust_ID),
CONSTRAINT comp_fk FOREIGN KEY(Comp_ID) REFERENCES CoID(Comp_ID),
CONSTRAINT Staff_id_fk FOREIGN KEY(StaffID) REFERENCES
ConsultantID(Consult_ID)
);

--Insert values to table CID
INSERT INTO CID VALUES ('Z0001', 'John Doe', 'Bachelor of Science', 'C0001', 'K1');
INSERT INTO CID VALUES ('Z0002', 'Alice Smith', 'Master of Business Administration',
'C0002', 'K2');
INSERT INTO CID VALUES ('Z0003', 'Bob Johnson', 'Bachelor of Arts', 'C0003', 'K3');
INSERT INTO CID VALUES ('Z0004', 'Emily Davis', 'Doctor of Medicine', 'C0004', 'K4');
INSERT INTO CID VALUES ('Z0005', 'Michael Wilson', 'Master of Science in Engineering',
'C0005', 'K5');
```

Customer

```
-- Customer
--Table to store Customer ID & Email
create table Customer(
Cust_ID varchar2 (5),
Cust_Email varchar2 (50),
CONSTRAINT C_pk PRIMARY KEY ( Cust_ID,Cust_Email ),
CONSTRAINT cufk FOREIGN KEY(Cust_ID) REFERENCES CID(Cust_ID),
CONSTRAINT custo_fk FOREIGN KEY(Cust_Email) REFERENCES CEmail(Cust_Email)
);

--Insert values into table CUser
INSERT INTO Customer VALUES ('Z0001', 'user1@example.com');
INSERT INTO Customer VALUES ('Z0002', 'john.doe@email.com');
INSERT INTO Customer(Cust_ID, Cust_Email) VALUES ('Z0003',
'alice.smith@email.com');
INSERT INTO Customer(Cust_ID, Cust_Email) VALUES ('Z0004', 'jdoe2022@email.com');
INSERT INTO Customer(Cust_ID, Cust_Email) VALUES ('Z0005',
'admin.user@email.com');
```

Cust_Contact

```
--Create customer contact with contact number and ID
Create table Cust_Contact(
   Cust_Contact_Num varchar(50),
   Cust_ID varchar(5),
   CONSTRAINT cont_pk PRIMARY KEY (Cust_Contact_Num),
   CONSTRAINT sl_fk FOREIGN KEY (Cust_ID) REFERENCES CID (Cust_ID)
);

--Insert the data of contact num and ID for customer
INSERT INTO Cust_Contact(Cust_Contact_Num, Cust_ID)
VALUES ('015 8765 432', 'Z0001');
INSERT INTO Cust_Contact(Cust_Contact_Num, Cust_ID)
VALUES ('018 2345 678', 'Z0002');
INSERT INTO Cust_Contact(Cust_Contact_Num, Cust_ID)
VALUES ('016 8765 432', 'Z0003');
INSERT INTO Cust_Contact(Cust_Contact_Num, Cust_ID)
VALUES ('010 5432 109', 'Z0004');
INSERT INTO Cust_Contact(Cust_Contact_Num, Cust_ID)
VALUES ('019 8765 432', 'Z0005');
```

Package

```
-- Package
--Table to store table package ID , Name , Price , Package description
CREATE TABLE Package
(
	Pack_ID VARCHAR2(2) PRIMARY KEY,
	Pack_Name VARCHAR2(50),
	Pack_Price NUMBER(4),
	Pack_Desc VARCHAR2(200)
);

--Insert values into table Package
INSERT INTO Package VALUES ('N1','CV Template',99,'2 CV Template');
INSERT INTO Package VALUES ('N2','Basic Writing',159,'5 Days Duration');
INSERT INTO Package VALUES ('N3','Pro Writing',199,'4 Days Duration');
```

Cust_Pack

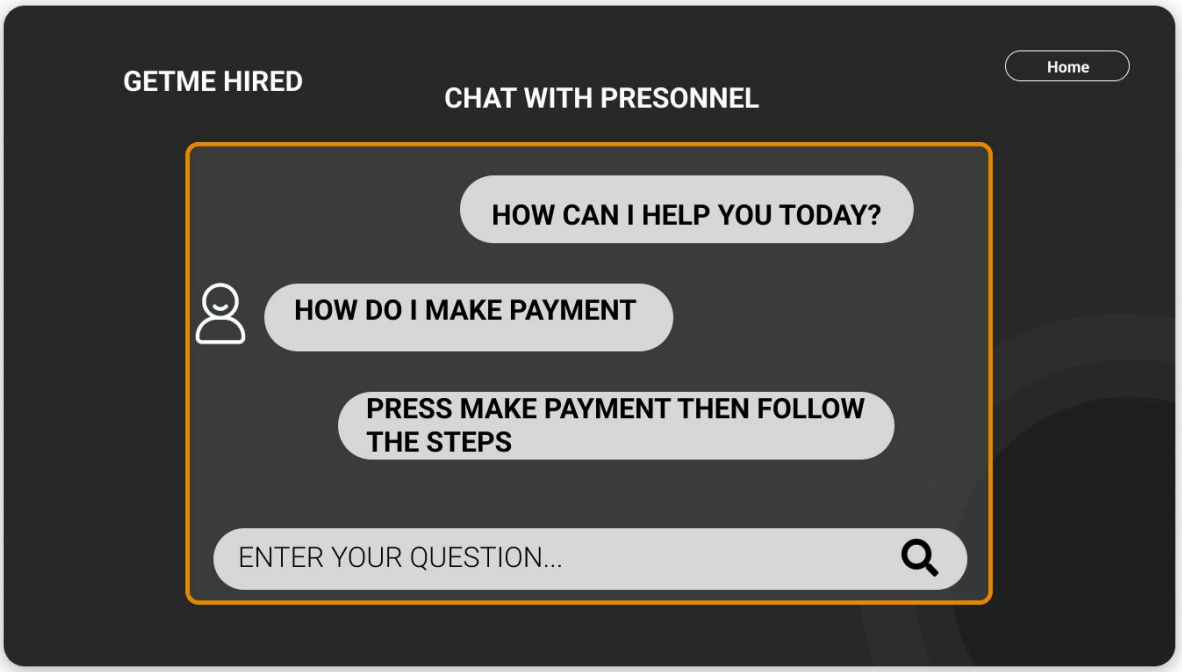
```
--Table to store Cust_Pack which has Customer ID , Package ID
CREATE TABLE Cust_Pack
(
	Cust_ID VARCHAR2(5),
	Pack_ID VARCHAR2(2),
	CONSTRAINT pc_pk PRIMARY KEY (Cust_ID,Pack_ID),
	CONSTRAINT pcp_fk FOREIGN KEY (Cust_ID) REFERENCES CID(Cust_ID),
	CONSTRAINT pcs_fk FOREIGN KEY (Pack_ID) REFERENCES
Package(Pack_ID)
);


--Insert values into table Cust_Pack
INSERT INTO Cust_Pack VALUES ('Z0001','N1');
INSERT INTO Cust_Pack VALUES ('Z0002','N1');
INSERT INTO Cust_Pack VALUES ('Z0002','N2');
INSERT INTO Cust_Pack VALUES ('Z0003','N3');
INSERT INTO Cust_Pack VALUES ('Z0004','N3');
```

8.0 Queries

1) Queries to see the Admin's Question and the answer of the question by combining them with Admin ID and their name.

```
SELECT ad.admin_id AS "Admin ID" , adm.Staff_Name AS "Admin Name"  ,
q.question_description AS "Question" , a.answer_description AS "Answer"

FROM AdminQA ad

JOIN AdminID adm ON (ad.admin_id = adm.Staff_ID)

JOIN Question q ON (q.question_id = ad.que_id)

JOIN Answer a ON (a.answer_id = ad.ans_id);
```

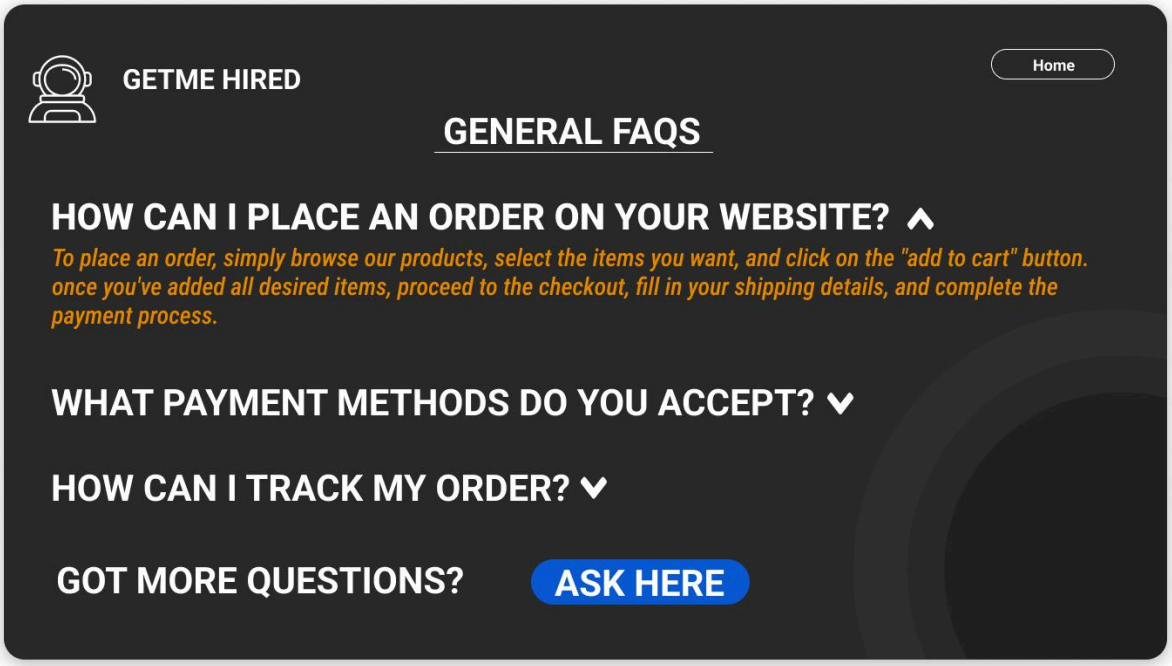
	Admin ID	Admin Name	Question	Answer
1	X0001	John Doe	What are the payment methods available	There are two payment methods available
2	X0002	Jane Smith	How can I customize my CV	We offer a variety of CV templates, including professional, modern, and creative styles
3	X0003	Mike Jones	Can I import my LinkedIn profile	You can customize your CV
4	X0004	Sarah White	How do I export my CV	You can import your LinkedIn profile to quickly create a CV
5	X0005	Robert Green	Can I track how many people viewed my CV	You can export your CV in PDF format



2) Queries to see the Question and the answer of the question by combining them with Questions' ID and their Answers' ID.

```
SELECT qaa.quest_id AS "Question Id" , q.question_description AS "Question" , qaa.ans_id AS "Answer Id" , a.answer_description AS "Answer"

FROM QA qaa

JOIN Question q ON (q.question_id = qaa.quest_id)

JOIN Answer a ON (a.answer_id = qaa.ans_id);
```

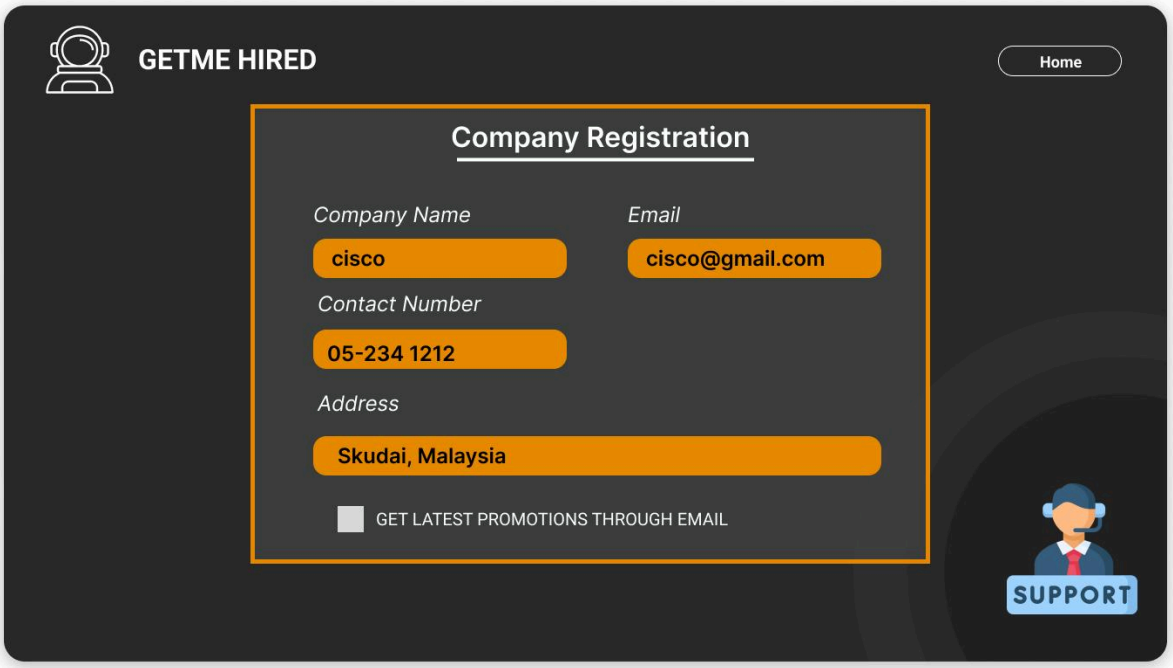
	Question Id	Question	Answer Id	Answer
1	Q0001	What are the payment methods available	A0001	There are two payment methods available
2	Q0002	How can I customize my CV	A0002	We offer a variety of CV templates, including professional, modern, and creative styles
3	Q0003	Can I import my LinkedIn profile	A0003	You can customize your CV
4	Q0004	How do I export my CV	A0004	You can import your LinkedIn profile to quickly create a CV
5	Q0005	Can I track how many people viewed my CV	A0005	You can export your CV in PDF format
6	Q0006	Is there a limit to how many CVs I can create	A0006	We provide a feature that allows you to track how many people viewed your CV
7	Q0007	Can I add a photo to my CV	A0007	There is no limit to how many CVs you can create
8	Q0008	How do I add references to my CV	A0008	You can add a photo to your CV
9	Q0009	Can I change the font and color scheme of my CV	A0009	You can add references to your CV in the references section
10	Q0010	How do I delete my CV	A0010	You can change the font and color scheme of your CV
11	Q0011	Can I share my CV on social media	A0011	You can delete your CV from the dashboard
12	Q0012	How do I add a cover letter to my CV	A0012	You can share your CV on social media platforms
13	Q0013	Can I save my CV as a PDF	A0013	You can add a cover letter to your CV in the cover letter section
14	Q0014	How do I add my educational background	A0014	You can save your CV as a PDF
15	Q0015	How do I add my work experience	A0015	You can add your educational background in the education section
16	Q0016	Can I add a skills section to my CV	A0016	You can add your work experience in the experience section
17	Q0017	How do I add languages to my CV	A0017	You can add a skills section to your CV
18	Q0018	Can I add certifications to my CV	A0018	You can add languages to your CV in the languages section
19	Q0019	How do I add volunteer work to my CV	A0019	You can add certifications to your CV
20	Q0020	Can I add a hobbies section to my CV	A0020	You can add volunteer work to your CV in the volunteer work section



3) Queries to see the company name, company email, contact number and their contact number by combining them with company ID.

```
SELECT ca.Comp_Name AS "Company Name" , ca.Comp_Email AS "Email" ,
cb.Comp_Contact_NUM AS "Contact Number" , ca.Address AS "Address"

FROM CoID ca

JOIN Comp_Contact cb ON (ca.Comp_ID = cb.Comp_ID);
```

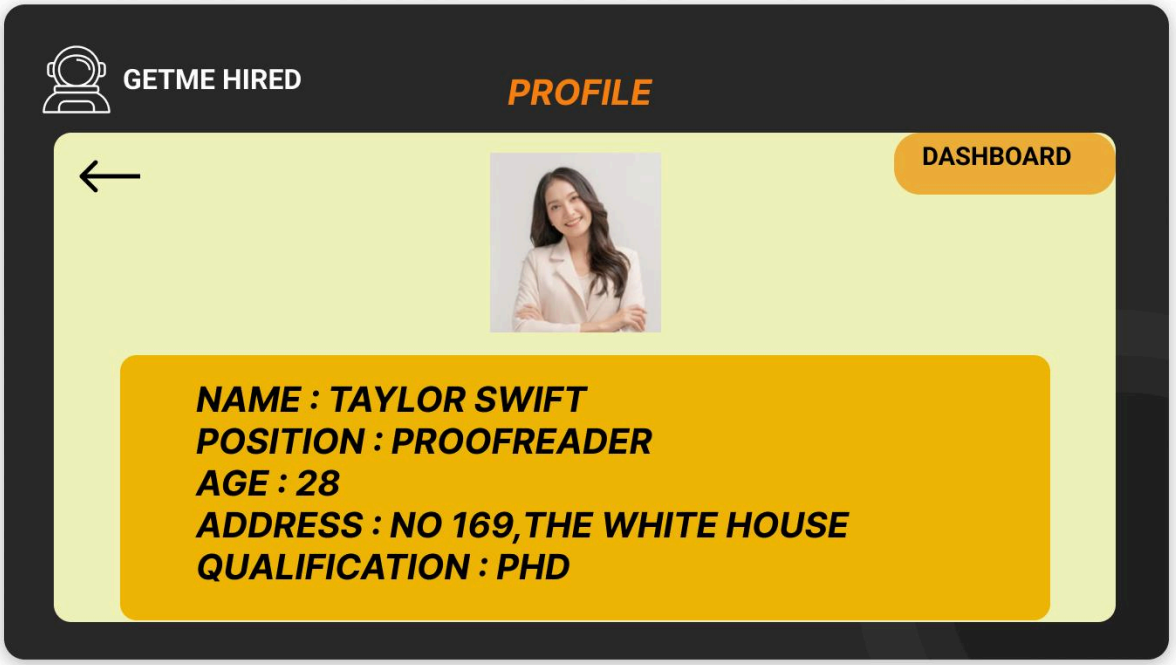
	Company Name	Email	Contact Number	Address
1	Tech Innovators Inc	techinnovators@example.com	012 456 7890	123 Tech Street, Tech City
2	Data Masters Ltd	datamasters@example.com	011 233 4455	456 Data Avenue, Data City
3	Cloud Tech Solutions	cloudtech@example.com	017 876 5432	789 Cloud Lane, Cloud City
4	Innova Systems LLC	innova@example.com	014 578 9012	101 Innovate Road, Innovate City
5	Rapid Systems Co	rapidsystems@example.com	013 875 4321	202 Rapid Street, Rapid City



4) Queries to see the proofreader ID, proofreader name, and their email by combining them with staffID of the proofreader relation.

```
SELECT pe.Staff_ID AS "Proofreader ID" , pi.Staff_Name AS "Proofreader Name" ,
pe.Staff_email AS "Email"

FROM StaffProof pe

JOIN ProofID pi ON (pe.Staff_ID = pi.Staff_ID);
```

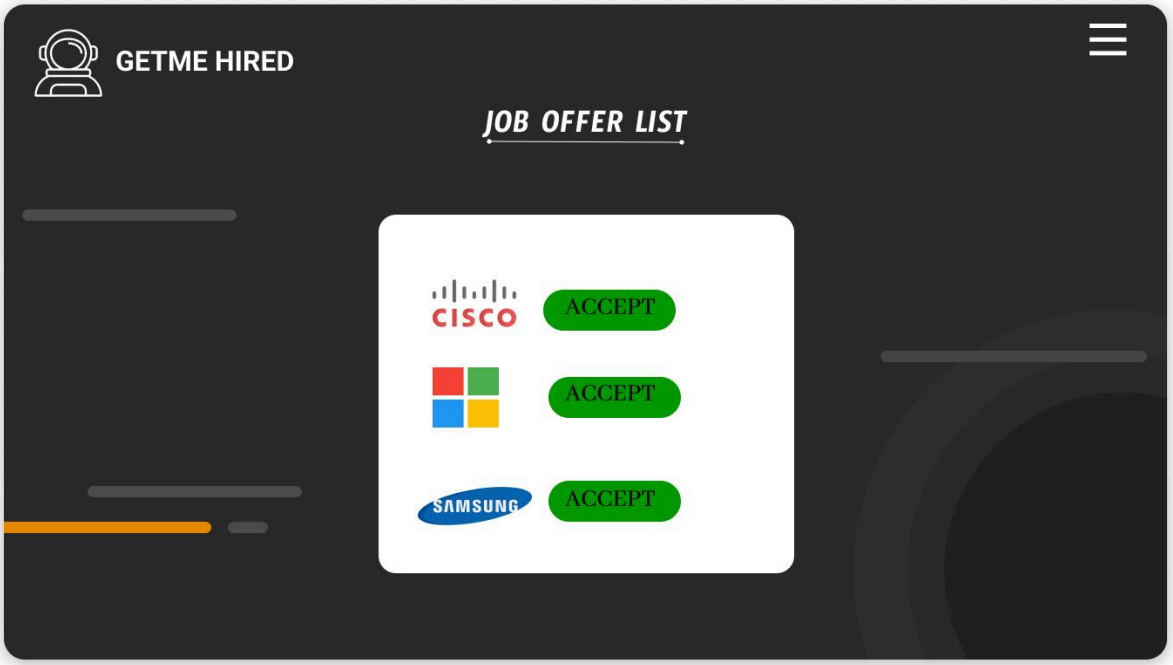
	Proofreader ID	Proffreader Name	Email
1	P0021	Alex Smith	alex.smith@example.com
2	P0022	Sophie Jones	sophie.jones@example.com
3	P0023	Michael Brown	michael.brown@example.com
4	P0024	Olivia Wilson	olivia.wilson@example.com
5	P0025	Ryan Taylor	ryan.taylor@example.com



5) Queries to see the customer name that have same company by combining them with company ID and by joining customer relation and company relation

```
SELECT a.Cust_Name AS "Customer Name" , b.Comp_Name
FROM CID a
JOIN COID b ON(a.Comp_ID = b.Comp_ID)
WHERE Comp_Name = 'Rapid Systems Co';
```

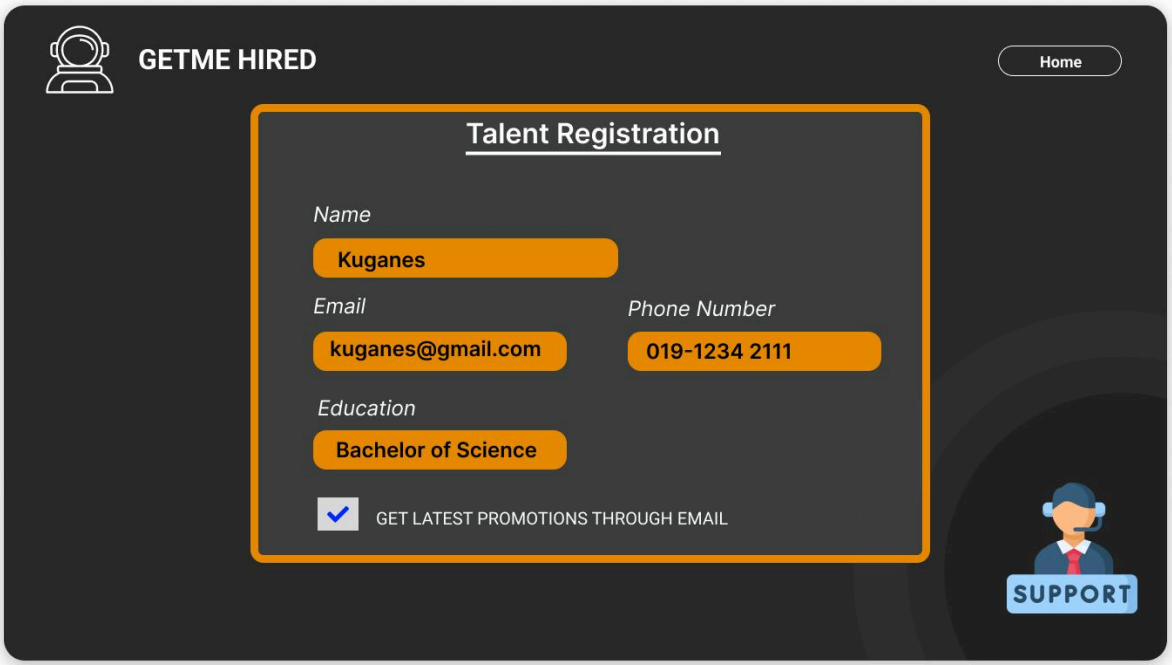
	Customer Name	COMP_NAME
1	Michael Wilson	Rapid Systems Co
2	Kuganes	Rapid Systems Co



6) Queries to join the customer name , customer email, contact number and education by company ID joining Customer, Cust_Contact and COID

```
SELECT ca.Cust_Name AS "Customer Name" , cb.Cust_Email AS "Customer Email",
cc.Cust_Contact_Num AS "Phone Number", ca.Education AS "Education", ca.Comp_ID,
compn.Comp_Name

FROM CID ca

JOIN Customer cb ON (ca.Cust_ID = cb.Cust_ID)

JOIN Cust_Contact cc ON (cc.Cust_ID = ca.Cust_ID)

JOIN COID compn ON (ca.Comp_ID = compn.Comp_ID);
```

	Customer Name	Customer Email	Phone Number	Education	COMP_ID	COMP_NAME
1	John Doe	user1@example.com	015 8765 432	Bachelor of Science	C0001	Tech Innovators Inc
2	Alice Smith	john.doe@email.com	018 2345 678	Master of Business Administration	C0002	Data Masters Ltd
3	Bob Johnson	alice.smith@email.com	016 8765 432	Bachelor of Arts	C0003	Cloud Tech Solutions
4	Emily Davis	jdoe2022@email.com	010 5432 109	Doctor of Medicine	C0004	Innova Systems LLC
5	Michael Wilson	admin.user@email.com	019 8765 432	Master of Science in Engineering	C0005	Rapid Systems Co



GETME HIRED

CUSTOMER DASHBOARD

CV | ASSIGNED CONSULTANT | ACCEPTED TIME | JOB OFFER

PROFILE

NAME : HAZIM BIN ALI
USERNAME : hazim
AGE : 24
ADDRESS : PortDIckson, Negeri Sembilan
EDUCATION : Diploma in Sceince

GETME HIRED

ADMIN DASHBOARD

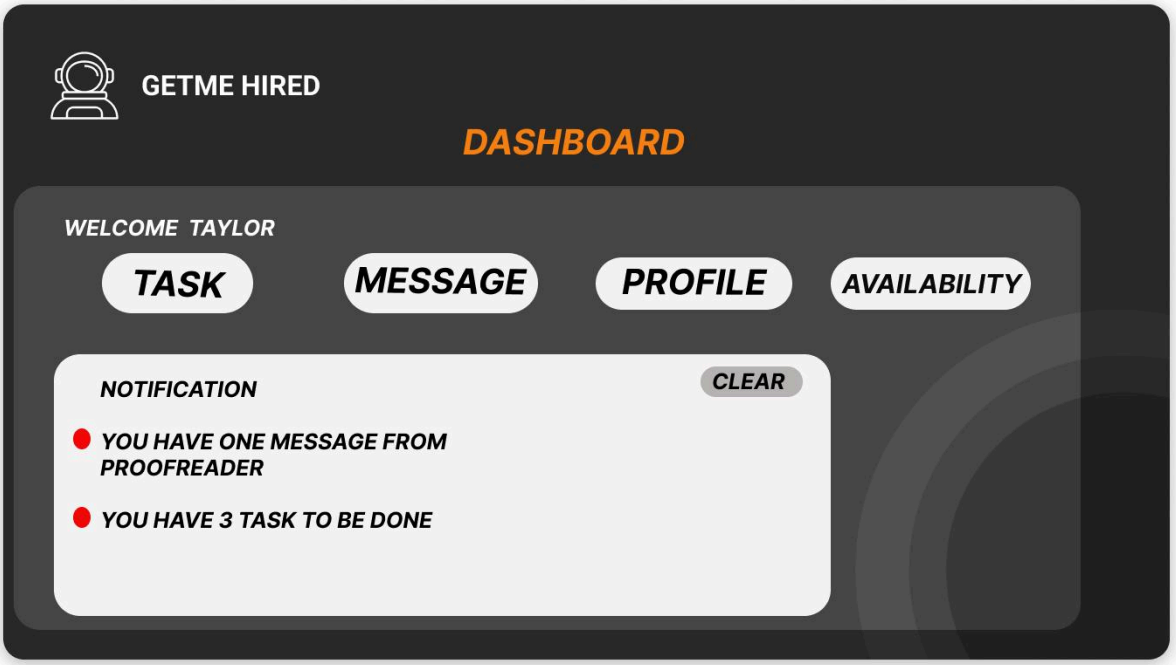
Welcome Hazinah

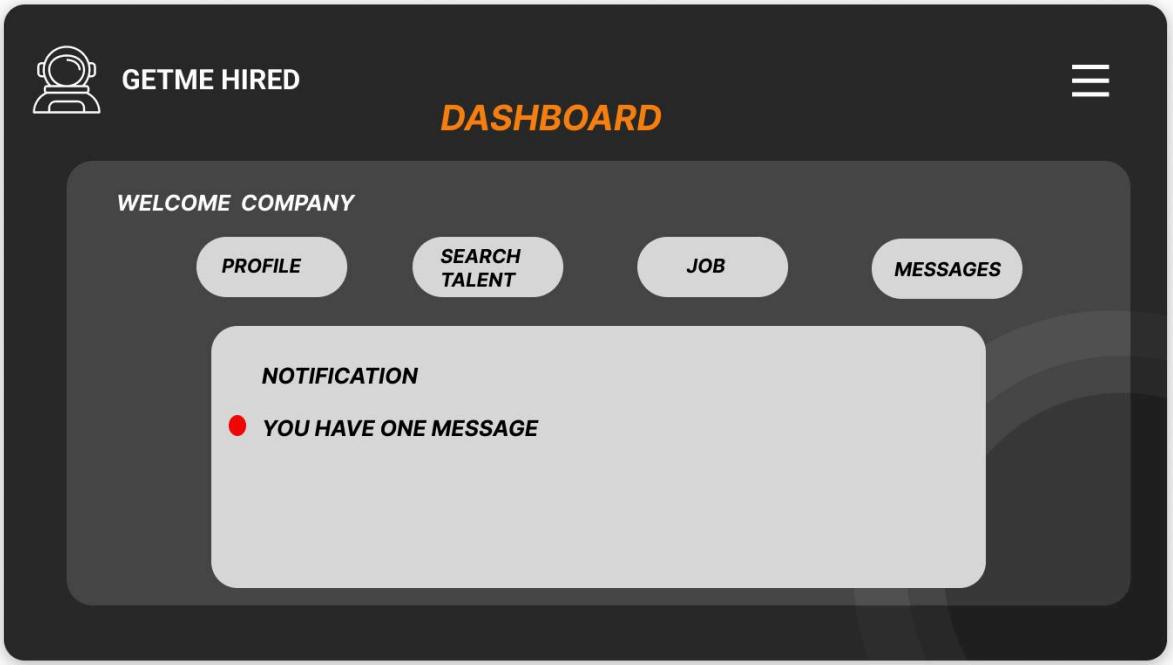
PROFILE

NAME : HAZINA BINTI ALI
USERNAME : hazinah
AGE : 40
ADDRESS : Johor Bahru
OFFICE LOCATION : Skudai

RECEIVED CV

QUERIES


GETME HIRED
DASHBOARD
WELCOME TAYLOR
TASK
MESSAGE
PROFILE
AVAILABILITY
NOTIFICATION
CLEAR
YOU HAVE ONE MESSAGE FROM PROOFREADER
YOU HAVE 3 TASK TO BE DONE


GETME HIRED
DASHBOARD
WELCOME COMPANY
PROFILE
SEARCH TALENT
JOB
MESSAGES
NOTIFICATION
YOU HAVE ONE MESSAGE

9.0 Figma Link

https://www.figma.com/file/zJSRQ5Ffg2fNwbEZ44FGyV/SAD-P2?type=design&node-id=538-18&mode=design&t=GGCw8y5NMflxvWT4-0

10.0 Summary

In this phase, Data Flow Diagram of the “To Be” system which has the context diagram , 0 diagram and the child diagrams was created. The proposed business rule , data requirements , transaction requirement , both conceptual and enhanced ERD and finally the data dictionary were identified. In this “To Be” system, all the new features to support the current system were added and the problem faced by the current system overcame.

Hence by completing this phase, the structure of the new system which has the automated CV template forwarding feature was identified, a platform for the admins to share information on a regular basis , a platform for communication between the admins and users , a library system for the companies to headhunt for talented workers and consultants to consult users.Thus this well structured system design plays an important role for the new system to work more efficiently than the current system and satisfy the customer more.

11.0 Reference

1) Van Nguyen, M., Min, B., Dernoncourt, F., & Nguyen, T. (2022, July). Joint extraction of entities, relations, and events via modeling inter-instance and inter-label dependencies. In Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (pp. 4363-4374).

2) Chang, H., Xu, H., van Genabith, J., Xiong, D., & Zan, H. (2023). JoinER-BART: Joint Entity and Relation Extraction with Constrained Decoding, Representation Reuse and Fusion. IEEE/ACM Transactions on Audio, Speech, and Language Processing.

3) Sehgal, S., Gupta, R. S., Wlodarski, M., Bilaver, L. A., Wehbe, F. H., Spergel, J. M., ... & Starren, J. B. (2022). Development of Food Allergy Data Dictionary: toward a food allergy data commons. The Journal of Allergy and Clinical Immunology: In Practice, 10(6), 1614-1621.