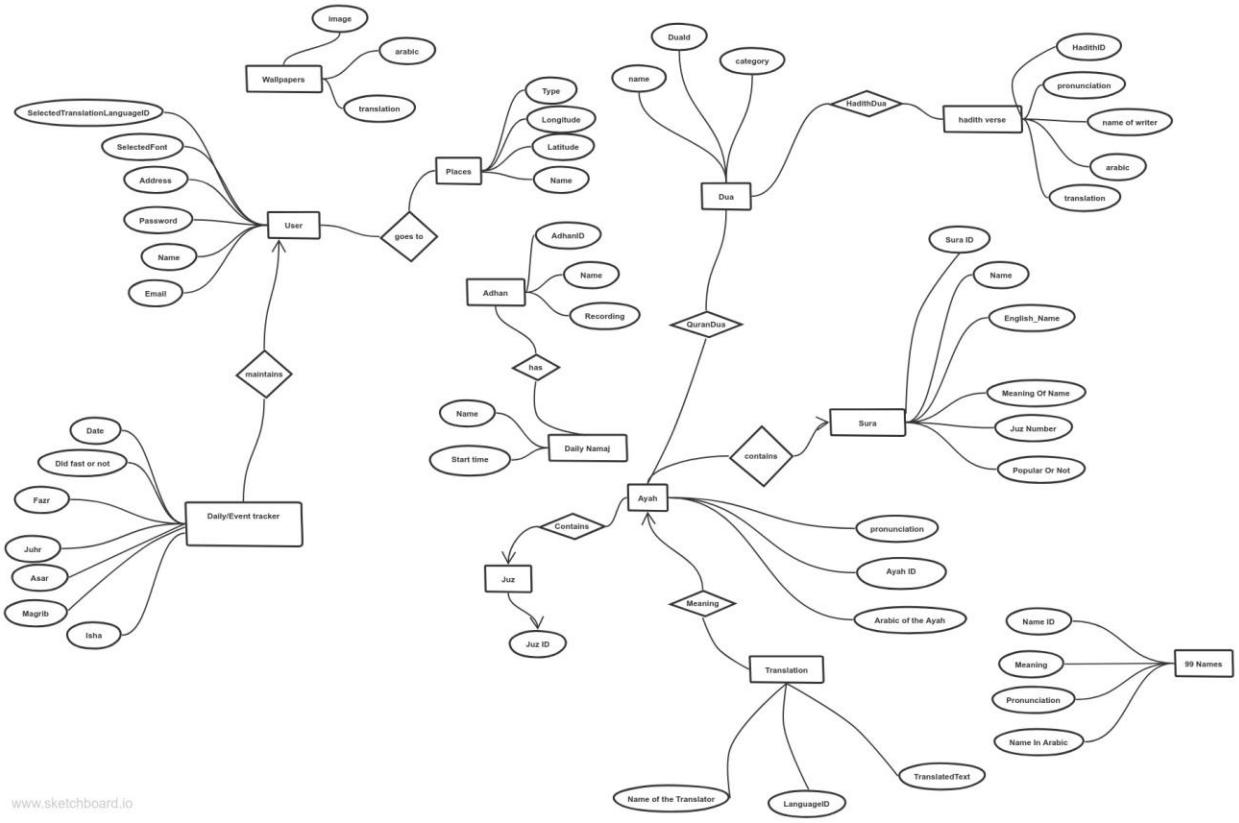
MUSLIM PRO

ERD :



Simple Queries:

1.Total Suras with their Arabic names,English names,meanings.

2.Origin of the sura.

3.Popular Sura

4.Searching Suras with the first alphabet of their names.
5.Pronunciation of a particular ayah.
6.5 times prayer times
7.Namaj Tracker(Searching by userID)
8.Fast Tracker
9.Hadith with Arabic and Translation
10.Particular Hadith books with Arabic and translation
11.Finding the source of a hadith
12.99 names of Allah(SWT)
13.Duas
14.Wallpaper messages

Complex Queries:

1.Full Sura with
Arabic,translation,pronunciation
searched by name.
2.Juz with
Arabic,pronunciation,translation
3.Translation of an ayah in selected language.
4.Translation of a Sura in selected language.
5.Dua from Quran.
6.Dua from Hadith.
7.Playing adhan.
8.All nearby places of an user.
9.Nearby Halal Places.
10.Nearby Mosques.

11.Finding the dates of an user's particular prayer
12.Namaj tracker searched by user's name.

FUNCTIONS:

1.Adds Dua From Ayah into the Dua table and updates Dua_ayah_relation Table.
2.Adds Dua From Hadithverse into the Dua table and updates Dua_hadith_relation Table.

3.Returns total number of suras of specific type.

TRIGGERS:

1.TRIGGER trig_trac_user AFTER INSERT ON usertable. This adds a new row on daily_event_tracker for the new user.

2.TRIGGER trig_hadith_dua AFTER update or insert ON dua_hadith_relation. This assigns the dua type 'Hadith' in dua table.

3.TRIGGER trig_place before insert ON usergoestoplaces.This updates place table if user goes to a new place.

SIMPLE QUERIES(CODE)

1.Total Suras with their Arabic names,English names,meanings.

```
SELECT SuraID,Name,English_Name,Meaning
FROM Sura;
```

2.Origin of the sura.

```
SELECT SuraID,Name,English_Name,Meaning
FROM Sura
WHERE Type='Meccan';
```

3.Popular Sura

```
SELECT SuraID,Name,English_Name,Meaning
FROM Sura
WHERE Popular_Or_Not=TRUE;
```

4.Searching Suras with the first alphabet of their names.

```
SELECT *
FROM sura
WHERE Sura.english_name LIKE 'A%'
order by suraid asc;
```

5.Pronunciation of a particular ayah.

```
SELECT pronunciation
FROM ayah
WHERE suraid=1 AND ayahid=6;
```

6.5 times prayer times

```
SELECT Name,StartTime
from DailyNamajTime
WHERE Date='2019-02-14';
```

7.Namaj Tracker(Searching by userID)

```
SELECT dDate,Fazr,Juhr,Asr,Magrib,Isha
FROM DailyEventTracker
WHERE UserID=1;
```

8.Fast Tracker

```
SELECT dDate,DidFastOrNot
FROM DailyEventTracker
WHERE UserID=1;
```

9.Hadith with Arabic and Translation

```
SELECT hadithid,arabictext,translation
FROM hadithverse
ORDER BY hadithid;
```

10.Particular Hadith books with Arabic

```
SELECT arabictext,translation
from hadithverse
where source like 'Bulugh al-Maram%';
```

11.Finding the source of a hadith

```
SELECT source
FROM hadithverse
WHERE hadithid=40000;
```

12.99 names of Allah(SWT)

```
SELECT NameID,Arabic,Pronunciation,Translation
FROM NamesOfAllah;
```

13.Duas

```
SELECT dua_name,category
FROM dua
GROUP BY dua_name,category
ORDER BY dua_name;
```

14.Wallpaper messages

```
SELECT wallpaper.arabic, wallpaper.translation
FROM wallpaper
WHERE wallpaperid=1;
```

COMPLEX QUERIES(CODE)

1.Full Sura with Arabic,translation,pronunciation searched by name.

```
SELECT Ayah.AyahID, Ayah.AyahText, Ayah.Pronunciation, Translation.Translated_Text
FROM Ayah,
        Translation
WHERE Ayah.SuraID = (SELECT SuraID
                FROM Sura
                WHERE English_Name = 'Al-Faatiha')
  AND Translation.SuraID = Ayah.SuraID
  and Translation.AyahID = Ayah.AyahID
  and translation.languageid = 'Bangla';
```

2.Juz with Arabic,pronunciation,translation

```
SELECT Ayah.* , translation.translated_text
FROM Ayah , translation
where (((ayah.suraid = (SELECT suraid FROM juz WHERE juzid = 3) AND ayah.ayahid
>=(SELECT ayahid from juz where juzid = 3)) or
        (ayah.suraid > (SELECT suraid FROM juz WHERE juzid = 3) and ayah.suraid <
(SELECT suraid FROM juz WHERE juzid = 4)) or
        (ayah.suraid = (SELECT suraid from juz where juzid = 4) AND ayah.ayahid < (select
ayahid from juz where juzid = 4))) )
        and ayah.suraid = translation.suraid and ayah.ayahid = translation.ayahid and
translation.languageid = 'Bangla'
ORDER BY (ayah.suraid, ayah.ayahid);
```

```
SELECT Ayah.* , translation.translated_text
FROM Ayah , translation
where (((ayah.suraid = (SELECT suraid FROM juz WHERE juzid = 30) AND ayah.ayahid
>=(SELECT ayahid from juz where juzid = 30)) or
        (ayah.suraid > (SELECT suraid FROM juz WHERE juzid = 30)))and ayah.suraid =
translation.suraid and ayah.ayahid = translation.ayahid and translation.languageid = 'Bangla')
ORDER BY (ayah.suraid, ayah.ayahid);
```

3.Translation of an ayah in selected language.

```
SELECT translated_text
FROM ayah,translation
WHERE ayah.suraid=1 AND ayah.ayahid=6 AND ayah.suraid=translation.suraid AND
ayah.ayahid=translation.ayahid
AND translation.name_of_translator='Muhiuddin Khan' AND translation.languageid='Bangla';
```

4.Translation of a Sura in selected language.

```
SELECT translated_text
FROM ayah,translation
WHERE ayah.suraid=1 AND ayah.suraid=translation.suraid AND
ayah.ayahid=translation.ayahid
AND translation.name_of_translator='Muhiuddin Khan' AND translation.languageid='Bangla';
```

5.Dua from Quran.

```
SELECT ayah.*, translation.translated_text
FROM Dua,
        ayah,
        dua_ayah_relation,
        translation
WHERE dua.DuaID = dua_ayah_relation.DuaID
  AND dua_ayah_relation.SuraID = ayah.SuraID
  AND dua_ayah_relation.AyahID = ayah.AyahID
  AND dua.category = 'Morning And Evening'
  AND dua.dua_name = 'When Waking Up'
  AND dua.dua_type = 'Ayah'
  AND translation.suraid = ayah.suraid
  AND translation.ayahid = ayah.ayahid
  AND translation.languageid = 'Bangla'
  AND translation.name_of_translator = 'Muhiuddin Khan';
```

6.Dua from Hadith.

```
SELECT hadithverse.*
FROM Dua,HadithVerse, dua_hadith_relation
WHERE dua.DuaID = dua_hadith_relation.DuaID
  AND dua_hadith_relation.HadithID = HadithVerse.HadithID
  AND dua.category = 'Demo1'
  AND dua.dua_name = 'HadithDua1'
  AND dua.dua_type = 'Hadith';
```

7.Playing adhan.

```
SELECT adhan.audiopath
FROM adhan,dailynamajadhanrelation,dailynamajtime
WHERE adhan.adhanid=dailynamajadhanrelation.adhanid AND
dailynamajadhanrelation.name=dailynamajtime.name AND
        adhan.adhanid=1 AND adhan.info='Adhan(Madina) Fajr' AND
dailynamajtime.name='Juhr' AND
to_char(dailynamajtime.starttime,'HH12:MI:SS')=to_char(now()::Time, 'HH12:MI:SS');
```

8.All nearby places of an user.

```
SELECT Place.Name,place.type
FROM Place,
        UserTable,
        UserGoesToPlaces
WHERE UserTable.UserID = UserGoesToPlaces.UserId
  AND UserGoesToPlaces.Longitude = Place.Longitude
  AND UserGoesToPlaces.Latitude = Place.Latitude
  AND UserTable.UserID = 1
  AND ((abs(UserTable.Latitude - Place.Latitude) <= 10 OR abs(Place.Latitude -
UserTable.Latitude) <= 10) AND
        (abs(UserTable.Longitude - Place.Longitude) <= 10 OR abs(Place.Longitude -
UserTable.Longitude) <= 10));
```

9.Nearby Halal Places

```
SELECT Place.Name
FROM Place,UserTable,UserGoesToPlaces
WHERE UserTable.UserID=UserGoesToPlaces.UserId AND
UserGoesToPlaces.Longitude=Place.Longitude AND
        UserGoesToPlaces.Latitude=Place.Latitude AND UserTable.UserID=1 AND
Place.Type='Halal Place' AND
        ((abs(UserTable.Latitude-Place.Latitude)<=1.5 OR abs(Place.Latitude-
UserTable.Latitude)<=1.5) AND
        (abs(UserTable.Longitude-Place.Longitude )<=1.5 OR abs(Place.Longitude-
UserTable.Longitude )<=1.5));
```

10.Nearby Mosques.

```
SELECT Place.Name
FROM Place,UserTable,UserGoesToPlaces
```

```
WHERE UserTable.UserID=UserGoesToPlaces.UserId AND
UserGoesToPlaces.Longitude=Place.Longitude AND
        UserGoesToPlaces.Latitude=Place.Latitude AND UserTable.UserID=1 AND
Place.Type='Mosque' AND
        ((abs(UserTable.Latitude-Place.Latitude)<=1.5 OR abs(Place.Latitude-
UserTable.Latitude)<=1.5) AND
        (abs(UserTable.Longitude-Place.Longitude )<=1.5 OR abs(Place.Longitude-
UserTable.Longitude )<=1.5));
```

11.Finding the dates of an user's particular prayer

```
SELECT ddate
FROM dailyeventtracker,usertable
WHERE dailyeventtracker.userid=usertable.userid AND dailyeventtracker.fazr=TRUE;
```

12.Namaj tracker searched by user's name.

```
SELECT ddate,fazr,juhr,asr,magrib,isha
FROM dailyeventtracker,usertable
WHERE dailyeventtracker.userid=usertable.userid AND usertable.name='Adiba';
```

FUNCTIONS(CODE)

1.Adds Dua From Ayah into the Dua table and updates Dua_ayah_relation Table.

```
CREATE OR REPLACE function func_add_ayah_dua(dID INTEGER, dNAME
VARCHAR(100), dCAT VARCHAR(100), sID INTEGER, aID INTEGER, type
VARCHAR(100)) returns varchar(1000)
 LANGUAGE plpgsql
AS
$$
BEGIN
 IF (sID IN (SELECT SuraID FROM Ayah) AND aID IN (SELECT AyahID FROM Ayah
WHERE SuraID = sID)) THEN
   INSERT INTO Dua
   VALUES (dID, dNAME, dCAT, type);

   INSERT INTO Dua_Ayah_Relation
   VALUES (dID, sID, aID);

 ELSE
   RAISE NOTICE 'PLEASE CHECK SURA ID AND AYAH ID';
 END IF;
```

```
 return 'Dua added in dua table and dua_ayah_relation table';
END;
$$;
```

2.Adds Dua From Hadithverse into the Dua table and updates Dua_hadith_relation Table.

```
CREATE OR REPLACE function func_add_hadith_dua(dID INTEGER, dNAME
VARCHAR(100), dCAT VARCHAR(100), hID INTEGER, type VARCHAR(100)) returns
varchar(1000)
 LANGUAGE plpgsql
AS
$$
BEGIN
 IF (hID IN (SELECT HadithID FROM HadithVerse)) THEN
   INSERT INTO Dua
   VALUES (dID, dNAME, dCAT, type);

   INSERT INTO Dua_Hadith_Relation
   VALUES (dID, hID);

 ELSE
   RAISE NOTICE 'PLEASE CHECK HADITH ID';
 END IF;
 return 'Hadith type dua added in dua and hadith_ayah_relation';
```

```
END;
$$;
```

3.Returns total number of suras of specific type.

```
CREATE OR REPLACE FUNCTION SuraOfType(hi VARCHAR(100))RETURNS
INTEGER
AS $$
 DECLARE
   R INTEGER;

   BEGIN

   SELECT COUNT(*) INTO R
   FROM sura
   WHERE sura.type =hi;

 RETURN R;



  END;
$$ LANGUAGE plpgsql;

SELECT SuraOfType('Meccan');
```

1. TRIGGER trig_trac_user AFTER INSERT ON usertable. This adds a new row on daily_event_tracker for the new user.

```
CREATE OR REPLACE FUNCTION trac_info()
 RETURNS trigger AS
$$
BEGIN
        INSERT INTO dailyeventtracker(ddate, userid, didfastornot, fazr, juhr, asr,
magrib, isha)
        VALUES(CURRENT_DATE, new.userid, false, false, false, false, false, false);

   RETURN NEW;
END;
$$
LANGUAGE 'plpgsql';


CREATE TRIGGER trig_trac_user
 AFTER INSERT
 ON usertable
 FOR EACH ROW
 EXECUTE PROCEDURE trac_info();
```

2.TRIGGER trig_hadith_dua AFTER update or insert ON dua_hadith_relation. This assigns the dua type 'Hadith' in dua table.

```
CREATE OR REPLACE FUNCTION trig_dua()
 RETURNS trigger AS
$$
BEGIN
   UPDATE dua
         set dua_type = 'Hadith'
         where duaid=new.duaid;

   delete from dua_ayah_relation
         where dua_ayah_relation.duaid = new.duaid;

   RETURN NEW;
END;
$$
LANGUAGE 'plpgsql';


CREATE TRIGGER trig_hadith_dua
 AFTER update or insert
 ON dua_hadith_relation
 FOR EACH ROW
 EXECUTE PROCEDURE trig_dua();
```

3.TRIGGER trig_place before insert ON usergoestoplaces.This updates place table if user goes to a new place.

```
CREATE OR REPLACE FUNCTION after_place_insert()
 RETURNS trigger AS
$$
BEGIN
   insert into place(longitude, latitude, name, type) VALUES (new.longitude,
new.latitude, 'place1', 'Halal Place');


   RETURN NEW;
END;
$$
LANGUAGE 'plpgsql';


CREATE TRIGGER trig_place
 before insert
 ON usergoestoplaces
 FOR EACH ROW
 EXECUTE PROCEDURE after_place_insert();
```

DDL

```
-- CREATING TABLES

CREATE OR REPLACE PROCEDURE create_tables()
  LANGUAGE plpgsql
AS
$$
BEGIN
  CREATE TABLE NamesOfAllah
  (
        NameID         NUMERIC PRIMARY KEY,
        Arabic         VARCHAR(100) NOT NULL,
        Pronunciation VARCHAR(100) NOT NULL,
        Translation   VARCHAR(100) NOT NULL
  );
  CREATE TABLE WallPaper
  (
        WallpaperID NUMERIC PRIMARY KEY,
        Arabic         VARCHAR(200) NOT NULL,
        ImagePath   VARCHAR(200) NOT NULL,
        Translation VARCHAR(200) NOT NULL
  );
```

```
CREATE TABLE UserTable
(
	UserID			NUMERIC PRIMARY KEY,
	Email			VARCHAR(100) NOT NULL,
	Password		VARCHAR(100) NOT NULL,
	Name			VARCHAR(100) NOT NULL,
	Longitude		FLOAT		NOT NULL,
	Latitude		FLOAT		NOT NULL,
	Address			VARCHAR(200) NOT NULL,
	SelectedFont		VARCHAR(100) NOT NULL,
	SelectedTranslationLanguage VARCHAR(100) NOT NULL
);
CREATE TABLE Place
(
	Longitude FLOAT	NOT NULL,
	Latitude  FLOAT	NOT NULL,
	Name  VARCHAR(100) NOT NULL,
	Type    VARCHAR(100) NOT NULL,
	PRIMARY KEY (Longitude, Latitude)
);
CREATE TABLE UserGoesToPlaces
(
	UserId NUMERIC NOT NULL,
	Longitude FLOAT   NOT NULL,
	Latitude  FLOAT   NOT NULL,
	PRIMARY KEY (UserId, Longitude, Latitude),
	FOREIGN KEY (UserId) REFERENCES UserTable (UserID),
	FOREIGN KEY (Longitude, Latitude) REFERENCES Place (Longitude, Latitude)
);
CREATE TABLE DailyEventTracker
(
	dDate		date				NOT NULL,
	UserID		NUMERIC REFERENCES UserTable (UserID) NOT NULL,
	DidFastOrNot BOOLEAN,
	Fazr		BOOLEAN,
	Juhr		BOOLEAN,
	Asr	BOOLEAN,
	Magrib		BOOLEAN,
	Isha		BOOLEAN,
	PRIMARY KEY (dDate, UserID)
);
CREATE TABLE Sura
(
	SuraID		NUMERIC PRIMARY KEY,
```

```
        Name            VARCHAR(100) NOT NULL,
        English_Name    VARCHAR(100) NOT NULL,
        Meaning         VARCHAR(100) NOT NULL,
        Type            VARCHAR(100) NOT NULL,
        Popular_Or_Not BOOLEAN
);

CREATE TABLE Ayah
(
        SuraID          NUMERIC REFERENCES Sura (SuraID) NOT NULL,
        AyahID          NUMERIC                          NOT NULL,
        AyahText        VARCHAR                          NOT NULL,
        Pronunciation VARCHAR,
        PRIMARY KEY (SuraID, AyahID)
);

CREATE TABLE Juz
(
        JuzID  NUMERIC NOT NULL PRIMARY KEY,
        SuraID NUMERIC NOT NULL,
        AyahID NUMERIC NOT NULL,
        FOREIGN KEY (SuraID, AyahID) REFERENCES Ayah (SuraID, AyahID)
);

CREATE TABLE HadithVerse
(
        HadithID        NUMERIC PRIMARY KEY,
        Source          VARCHAR,
        Arabic_Of_Hadith VARCHAR,
        Translation     VARCHAR
);
CREATE TABLE Dua
(
        DuaID  NUMERIC PRIMARY KEY,
        Dua_Name VARCHAR(100) NOT NULL,
        Category VARCHAR(100) NOT NULL,
        Dua_type VARCHAR(100) NOT NULL
);
CREATE TABLE Dua_Ayah_Relation
(
        DuaID  NUMERIC NOT NULL PRIMARY KEY,
        SuraID NUMERIC NOT NULL,
        AyahID NUMERIC NOT NULL,
        FOREIGN KEY (DuaID) REFERENCES Dua (DuaID),
```

```
        FOREIGN KEY (SuraID, AyahID) REFERENCES Ayah (SuraID, AyahID)
 );
 CREATE TABLE Dua_Hadith_Relation
 (
        DuaID  NUMERIC REFERENCES Dua (DuaID)                NOT NULL PRIMARY KEY,
        HadithID NUMERIC REFERENCES HadithVerse (HadithID) NOT NULL
 );
 CREATE TABLE Translation
 (
        LanguageID              VARCHAR(20)  NOT NULL,
        Name_Of_Translator VARCHAR(100) NOT NULL,
        Translated_Text         VARCHAR      NOT NULL,
        SuraID            NUMERIC      NOT NULL,
        AyahID            NUMERIC      NOT NULL,
        PRIMARY KEY (LanguageID, Name_Of_Translator, SuraID, AyahID),
        FOREIGN KEY (SuraID, AyahID) REFERENCES Ayah (SuraID, AyahID)
 );
 CREATE TABLE DailyNamajTime
 (
        Name  VARCHAR(100) PRIMARY KEY,
        StartTime TIME
 );
 CREATE TABLE Adhan
 (
        AdhanID   NUMERIC PRIMARY KEY,
        Info      VARCHAR(100) NOT NULL,
        AudioPath VARCHAR(200) NOT NULL
 );
 CREATE TABLE DailyNamajAdhanRelation
 (
        Name  VARCHAR(100) NOT NULL,
        AdhanID NUMERIC    NOT NULL,
        FOREIGN KEY (Name) REFERENCES DailyNamajTime (Name),
        FOREIGN KEY (AdhanID) REFERENCES Adhan (AdhanID),
        PRIMARY KEY (Name, AdhanID)
 );


  COMMIT;
END;
$$;


--Procedures
```

```
CREATE OR REPLACE PROCEDURE init_99names()
  LANGUAGE plpgsql
AS
$$
BEGIN
  COPY NamesOfAllah (nameid, arabic, pronunciation, translation) FROM 'F:\DATABASE
TERM PROJECT 2-2\Shokol csv excel file\names99.csv' DELIMITER ',' CSV HEADER;
  COMMIT;
END;
$$;

CREATE OR REPLACE PROCEDURE init_wallpaper()
  LANGUAGE plpgsql
AS
$$
BEGIN
  INSERT INTO WallPaper(WallpaperID, Arabic, Translation, ImagePath)
  VALUES (1, 'أحسن الحُسن حُسن الأدب', 'The best beauty, is beautiful manners',
        'F:\DATABASE TERM PROJECT 2-2\Images for wallpaper\img1.jpg');

  INSERT INTO WallPaper(WallpaperID, Arabic, Translation, ImagePath)
  VALUES (2, 'الافعال ابلغ من الاقوال', 'Actions speak louder ( are better or more eloquent or efficient)
than words .',
        'F:\DATABASE TERM PROJECT 2-2\Images for wallpaper\img2.jpg');

  INSERT INTO WallPaper(WallpaperID, Arabic, Translation, ImagePath)
  VALUES (3, 'من طلب العلا سهر الليالي',
        'He who wants to accomplish great things (or succeed) has to stay awake many nights
(or study/work hard) .',
        'F:\DATABASE TERM PROJECT 2-2\Images for wallpaper\img3.jpg');

  INSERT INTO WallPaper(WallpaperID, Arabic, Translation, ImagePath)
  VALUES (4, 'اتق شر الحليم اذا غضب', 'Beware the level-headed (calm/patient) person if they get
angry .',
        'F:\DATABASE TERM PROJECT 2-2\Images for wallpaper\img4.jpg');

  INSERT INTO WallPaper(WallpaperID, Arabic, Translation, ImagePath)
  VALUES (5, 'اختر أهون الشرين', 'Go with ( pick/choose) the lesser of two evils .',
        'F:\DATABASE TERM PROJECT 2-2\Images for wallpaper\img5.jpg');


  COMMIT;
END;
```

```
$$;

CREATE OR REPLACE PROCEDURE init_userTable()
  LANGUAGE plpgsql
AS
$$
BEGIN
  COPY UserTable (UserID, Email, Password, Name, Longitude, Latitude, Address,
SelectedFont,
                  SelectedTranslationLanguage) FROM 'F:\DATABASE TERM PROJECT 2-
2\Shokol csv excel file\UserTable1.csv' DELIMITER ',' CSV HEADER;
  COMMIT;
END;
$$;

CREATE OR REPLACE PROCEDURE init_place()
  LANGUAGE plpgsql
AS
$$
BEGIN
  COPY Place (Longitude, Latitude, Name, Type) FROM 'F:\DATABASE TERM PROJECT 2-
2\Shokol csv excel file\Place.csv' DELIMITER ',' CSV HEADER;
  COMMIT;
END;
$$;

CREATE OR REPLACE PROCEDURE init_UserGoesPlace()
  LANGUAGE plpgsql
AS
$$
BEGIN
  INSERT INTO UserGoesToPlaces(UserId, Longitude, Latitude)
  SELECT UserID,Place.Longitude,Place.Latitude
  FROM UserTable,
        Place;
  COMMIT;
END;
$$;

CREATE OR REPLACE PROCEDURE init_dailyEventTracker()
  LANGUAGE plpgsql
AS
$$
BEGIN
```

```
  COPY DailyEventTracker (dDate, UserID, DidFastOrNot, Fazr, Juhr, Asr, Magrib, Isha) FROM
'F:\DATABASE TERM PROJECT 2-2\Shokol csv excel file\UserDailyTracker.csv' DELIMITER ','
CSV HEADER;
  COMMIT;
END;
$$;

CREATE OR REPLACE PROCEDURE init_sura()
  LANGUAGE plpgsql
AS
$$
BEGIN
  COPY Sura (suraID, Name, english_Name, meaning, type) FROM 'F:\DATABASE TERM
PROJECT 2-2\Shokol csv excel file\surasFinal.csv' DELIMITER ',' CSV HEADER;
  COMMIT;
END;
$$;

create or replace procedure init_ayah()
  language plpgsql
as
$$
BEGIN
  CREATE TEMP TABLE temp_ayah
  (
        Database NUMERIC NOT NULL,
        SuraID   INTEGER NOT NULL,
        AyahID   NUMERIC NOT NULL,
        AyahText VARCHAR NOT NULL,
        PRIMARY KEY (SuraID, AyahID)
  ) ON COMMIT DROP;
  copy temp_ayah (Database, SuraID, AyahID, AyahText) FROM 'F:\DATABASE TERM
PROJECT 2-2\Shokol csv excel file\Arabic-(Original-Book)-1.csv' DELIMITER ',' CSV HEADER;

  CREATE TEMP TABLE PronunciationTest
  (
        SuraID NUMERIC NOT NULL,
        AyahID NUMERIC NOT NULL,
        Pronun VARCHAR NOT NULL
  ) ON COMMIT DROP;
  COPY PronunciationTest (SuraID, AyahID, Pronun) FROM 'F:\DATABASE TERM PROJECT
2-2\Shokol csv excel file\Pronunciation.csv' DELIMITER ',' CSV HEADER;
```

```
  INSERT INTO Ayah SELECT SuraID, AyahID, AyahText FROM temp_ayah ON CONFLICT
DO NOTHING;
  UPDATE Ayah
  SET pronunciation = Pronunciationtest.pronun
  FROM Pronunciationtest
  WHERE Pronunciationtest.suraid = Ayah.suraid
          AND Pronunciationtest.ayahid = Ayah.ayahid;


  COMMIT;
END;
$$;

CREATE OR REPLACE PROCEDURE init_juz()
  LANGUAGE plpgsql
AS
$$
BEGIN
  COPY Juz (juzid, suraid, ayahid) FROM 'F:\DATABASE TERM PROJECT 2-2\Shokol csv
excel file\juz.csv' DELIMITER ',' CSV HEADER;
  COMMIT;
END;
$$;

CREATE OR REPLACE PROCEDURE init_hadithVerse()
  LANGUAGE plpgsql
AS
$$
BEGIN
  COPY HadithVerse (hadithid,ArabicText,translation, source) FROM 'F:\DATABASE TERM
PROJECT 2-2\Shokol csv excel file\hadiths.csv' DELIMITER ',' CSV HEADER;
  COMMIT;
END;
$$;

CREATE OR REPLACE PROCEDURE add_ayah_dua(dID INTEGER, dNAME
VARCHAR(100), dCAT VARCHAR(100), sID INTEGER, aID INTEGER,
                                    type VARCHAR(100))
  LANGUAGE plpgsql
AS
$$
BEGIN
  IF (sID IN (SELECT SuraID FROM Ayah) AND aID IN (SELECT AyahID FROM Ayah WHERE
SuraID = sID)) THEN
```

```
        INSERT INTO Dua
        VALUES (dID, dNAME, dCAT, type);

        INSERT INTO Dua_Ayah_Relation
        VALUES (dID, sID, aID);

  ELSE
        RAISE NOTICE 'PLEASE CHECK SURA ID AND AYAH ID';
  END IF;
  COMMIT;
END;
$$;

CREATE OR REPLACE PROCEDURE add_hadith_dua(dID INTEGER, dNAME
VARCHAR(100), dCAT VARCHAR(100), hID INTEGER,
                                    type VARCHAR(100))
  LANGUAGE plpgsql
AS
$$
BEGIN
  IF (hID IN (SELECT HadithID FROM HadithVerse)) THEN
        INSERT INTO Dua
        VALUES (dID, dNAME, dCAT, type);

        INSERT INTO Dua_Hadith_Relation
        VALUES (dID, hID);

  ELSE
        RAISE NOTICE 'PLEASE CHECK HADITH ID';
  END IF;
  COMMIT;
END;
$$;

CREATE OR REPLACE PROCEDURE init_dua()
  LANGUAGE plpgsql
AS
$$
BEGIN
  CALL add_ayah_dua(1, 'When Waking Up', 'Morning And Evening', 3, 191, 'Ayah');-- THIS
WORKED FINE
  CALL add_ayah_dua(2, 'When Waking Up', 'Morning And Evening', 3, 192, 'Ayah');
  CALL add_ayah_dua(3, 'When Waking Up', 'Morning And Evening', 3, 193, 'Ayah');
```

```
  CALL add_ayah_dua(4, 'Remembrance In The Morning And Evening', 'Morning And Evening',
112, 2, 'Ayah');
  CALL add_ayah_dua(5, 'Remembrance In The Morning And Evening', 'Morning And Evening',
112, 3, 'Ayah');
  CALL add_ayah_dua(6, 'Remembrance In The Morning And Evening', 'Morning And Evening',
112, 4, 'Ayah');
  CALL add_ayah_dua(7, 'For Seeking Guidance', 'Praising Allah', 3, 160, 'Ayah');

  CALL add_hadith_dua(10, 'HadithDua1', 'Demo1', 1, 'Hadith');
  CALL add_hadith_dua(11, 'HadithDua2', 'Demo2', 15, 'Hadith');
  CALL add_hadith_dua(12, 'HadithDua3', 'Demo3', 100, 'Hadith');


  COMMIT;
END;
$$;

CREATE OR REPLACE PROCEDURE init_translation()
  LANGUAGE plpgsql
AS
$$
BEGIN

  CREATE TEMP TABLE BanglaTranslation
  (
        Database   INTEGER,
        SuraID          INTEGER NOT NULL,
        AyahID          INTEGER NOT NULL,
        BanglaText VARCHAR

  ) ON COMMIT DROP;

  COPY BanglaTranslation (Database, SuraID, AyahID, BanglaText) FROM 'F:\DATABASE
TERM PROJECT 2-2\Shokol csv excel file\Bangla-50.csv' DELIMITER ',' CSV HEADER;

  ALTER TABLE Translation
        ALTER COLUMN LanguageID
        SET DEFAULT 'Bangla';

  ALTER TABLE Translation
        ALTER COLUMN Name_Of_Translator
        SET DEFAULT 'Muhiuddin Khan';

  INSERT INTO Translation(SuraID, AyahID, Translated_Text)
```

```
SELECT SuraID,AyahID,BanglaText
FROM BanglaTranslation;

CREATE TEMP TABLE EnglishTranslationofAhmedAli
(
        Database        INTEGER,
        SuraID          INTEGER NOT NULL,
        AyahID          INTEGER NOT NULL,
        EnglishText1 VARCHAR

) ON COMMIT DROP;

COPY EnglishTranslationofAhmedAli (Database, SuraID, AyahID, EnglishText1) FROM
'F:\DATABASE TERM PROJECT 2-2\Shokol csv excel file\English-Ahmed-Ali-100.csv'
DELIMITER ',' CSV HEADER;

ALTER TABLE Translation
        ALTER COLUMN LanguageID
        SET DEFAULT 'English';

ALTER TABLE Translation
        ALTER COLUMN Name_Of_Translator
        SET DEFAULT 'AhmedAli';

INSERT INTO Translation(SuraID, AyahID, Translated_Text)
SELECT SuraID,AyahID,EnglishText1
FROM EnglishTranslationofAhmedAli;

CREATE TEMP TABLE UrduTranslationAhmedAli
(
        Database INTEGER,
        SuraID   INTEGER NOT NULL,
        AyahID   INTEGER NOT NULL,
        UrduText VARCHAR

) ON COMMIT DROP;


COPY UrduTranslationAhmedAli (Database, SuraID, AyahID, UrduText) FROM
'F:\DATABASE TERM PROJECT 2-2\Shokol csv excel file\Urdu.csv' DELIMITER ',' CSV
HEADER;

ALTER TABLE Translation
        ALTER COLUMN LanguageID
```

```
        SET DEFAULT 'Urdu';

  ALTER TABLE Translation
        ALTER COLUMN Name_Of_Translator
        SET DEFAULT 'Ahmed Ali';

  INSERT INTO Translation(SuraID, AyahID, Translated_Text)
  SELECT SuraID,AyahID,UrduText
  FROM UrduTranslationAhmedAli;

  COMMIT;
END;
$$;

CREATE OR REPLACE PROCEDURE init_dailynamajtime()
  LANGUAGE plpgsql
AS
$$
BEGIN
  COPY DailyNamajTime (Name, StartTime) FROM 'F:\DATABASE TERM PROJECT 2-
2\Shokol csv excel file\DailyNamazTime.csv' DELIMITER ',' CSV HEADER;
  COMMIT;
END;
$$;

CREATE OR REPLACE PROCEDURE init_adhan()
  LANGUAGE plpgsql
AS
$$
BEGIN
  COPY Adhan (AdhanID, Info, AudioPath) FROM 'F:\DATABASE TERM PROJECT 2-2\Shokol
csv excel file\Adhan er csv.csv' DELIMITER ',' CSV HEADER;
  COMMIT;
END;
$$;

CREATE OR REPLACE PROCEDURE init_dailynamajadhanrelation()
  LANGUAGE plpgsql
AS
$$
BEGIN
  INSERT INTO DailyNamajAdhanRelation(Name, AdhanID)
  SELECT Name,AdhanID
  FROM DailyNamajTime,
```

```
        Adhan;
  COMMIT;
END;
$$;

CREATE OR REPLACE PROCEDURE initialize()
  LANGUAGE plpgsql
AS
$$
BEGIN
  CALL init_99names();
  CALL init_wallpaper();
  CALL init_userTable();
  CALL init_place();
  CALL init_usergoesplace();
  CALL init_dailyEventTracker();
  CALL init_sura();
  CALL init_ayah();
  CALL init_juz();
  CALL init_hadithverse();
  CALL init_dua();
  CALL init_translation();
  CALL init_dailynamajtime();
  CALL init_adhan();
  CALL init_dailynamajadhanrelation();
  commit;
end;
$$;

CREATE OR REPLACE PROCEDURE drop_all()
  LANGUAGE plpgsql
AS
$$
BEGIN
  RAISE NOTICE 'Lets drop all the tables and procedures';

  DROP PROCEDURE init_99names();
  DROP PROCEDURE init_wallpaper();
  DROP PROCEDURE init_userTable();
  DROP PROCEDURE init_place();
  DROP PROCEDURE init_usergoesplace();
  DROP PROCEDURE init_dailyEventTracker();
  DROP PROCEDURE init_sura();
  DROP PROCEDURE init_ayah();
```

```
  DROP PROCEDURE init_juz();
  DROP PROCEDURE init_hadithverse();
  DROP PROCEDURE init_dua();
  DROP PROCEDURE init_translation();
  DROP PROCEDURE init_dailynamajtime();
  DROP PROCEDURE init_adhan();
  DROP PROCEDURE init_dailynamajadhanrelation();

  DROP PROCEDURE add_ayah_dua(dID INTEGER, dNAME VARCHAR, dCAT VARCHAR,
sID INTEGER, aID INTEGER, type VARCHAR(100));
  DROP PROCEDURE add_hadith_dua(dID INTEGER, dNAME VARCHAR, dCAT VARCHAR,
hID INTEGER, type VARCHAR(100));


  DROP TABLE DailyNamajAdhanRelation;
  DROP TABLE Adhan;
  DROP TABLE DailyNamajTime;
  DROP TABLE Translation;
  DROP TABLE Dua_Hadith_Relation;
  DROP TABLE Dua_Ayah_Relation;
  DROP TABLE Dua;
  DROP TABLE HadithVerse;
  DROP TABLE Juz;
  DROP TABLE Ayah;
  DROP TABLE Sura;
  DROP TABLE DailyEventTracker;
  DROP TABLE UserGoesToPlaces;
  DROP TABLE Place;
  DROP TABLE UserTable;
  DROP TABLE WallPaper;
  DROP TABLE NamesOfAllah;
  COMMIT;
END;
$$;


CALL create_tables();
CALL initialize();
CALL drop_all();
```