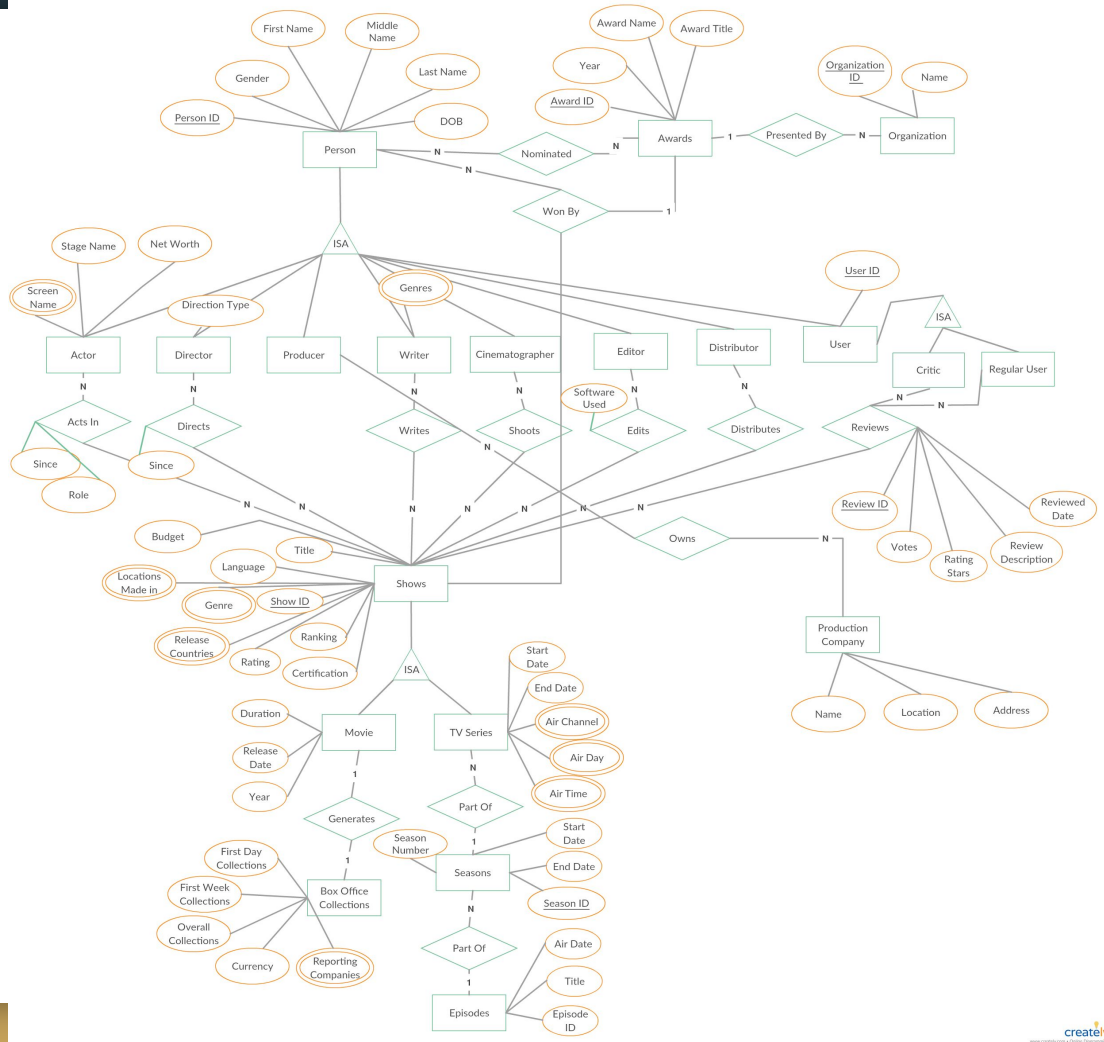
Movie Database

Jahnavi Chowdary Tharigopula, 112078393
Vamsikrishna Kodumuru Meesala, 112006882

E-R


First Name
Middle Name
Last Name
Gender
Person ID
DOB
Person
Award Name
Award Title
Year
Award ID
Awards
Organization ID
Name
Organization
Presented By
Nominated
Won By
ISA
Stage Name
Net Worth
Screen Name
Direction Type
Genres
Actor
Director
Producer
Writer
Cinematographer
Editor
Distributor
User
User ID
ISA
Critic
Regular User
Acts In
Directs
Writes
Shoots
Software Used
Edits
Distributes
Reviews
Since
Role
Since
Budget
Title
Language
Locations Made in
Genre
Show ID
Release Countries
Rating
Ranking
Certification
Shows
Owns
Review ID
Votes
Rating Stars
Review Description
Reviewed Date
Production Company
Name
Location
Address
ISA
Duration
Release Date
Year
Movie
TV Series
Start Date
End Date
Air Channel
Air Day
Air Time
Generates
Part Of
Season Number
Seasons
Start Date
End Date
Season ID
First Day Collections
First Week Collections
Overall Collections
Currency
Box Office Collections
Reporting Companies
Part Of
Episodes
Air Date
Title
Episode ID

Person

```
CREATE TABLE Person(
      Person_Id INTEGER NOT NULL AUTO_INCREMENT,
      Gender char(1) NOT NULL,
      First_Name VARCHAR(30) NOT NULL,
      Middle_Name VARCHAR(30),
      Last_Name VARCHAR(30) NOT NULL,
      DOB DATE NOT NULL,
      PRIMARY KEY(Person_Id),
      UNIQUE(First_Name,Last_Name,DOB,Gender),
      CHECK (Gender IN('M','F','O'))
);
```

Shows

```
CREATE TABLE Shows(
      Show_Id INTEGER NOT NULL AUTO_INCREMENT,
      Title VARCHAR(30),
      Language VARCHAR(30),
      Ranking INTEGER,
      Rating FLOAT,
      Certification VARCHAR (10),
      Budget INTEGER,
      Release_Countries VARCHAR(30),
      Locations_Made_In VARCHAR(30),
      PRIMARY KEY(Show_Id)
);
```

Actor

```
CREATE TABLE Actor(
	Person_Id INTEGER NOT NULL,
	Screen_Name VARCHAR(30),
	Stage_Name VARCHAR(30),
	Net_Worth INTEGER,
	Since_Year INTEGER,
	PRIMARY KEY(Person_Id),
	FOREIGN KEY(Person_Id) REFERENCES Person(Person_Id) ON DELETE CASCADE
);


CREATE TABLE Acting(
	Actor_Id INTEGER NOT NULL,
	Show_Id INTEGER NOT NULL,
	Role_First_Name VARCHAR(50),
	Role_Last_Name VARCHAR(50),
	PRIMARY KEY(Actor_Id,Show_Id),
	FOREIGN KEY(Actor_Id) REFERENCES Actor(Person_Id) ON DELETE CASCADE,
	FOREIGN KEY(Show_Id) REFERENCES Shows(Show_Id) ON DELETE CASCADE
);
```

Director

```
CREATE TABLE Director(
    Person_Id INTEGER NOT NULL,
    Direction_Type VARCHAR(10),
    Since_Year INTEGER,
    PRIMARY KEY(Person_Id),
    CHECK (Direction_Type IN('Music','Movie','Dance','Art')),
    FOREIGN KEY(Person_Id) REFERENCES Person(Person_Id) ON DELETE CASCADE
);
```

```
CREATE TABLE Direction(
    Director_Id INTEGER NOT NULL,
    Show_Id INTEGER NOT NULL,
    PRIMARY KEY(Director_Id,Show_Id),
    FOREIGN KEY(Director_Id) REFERENCES Director(Person_Id) ON DELETE CASCADE,
    FOREIGN KEY(Show_Id) REFERENCES Shows(Show_Id) ON DELETE CASCADE
);
```

Producer

```
/* Producer Table */

CREATE TABLE Producer(
    Person_Id INTEGER NOT NULL,
    PRIMARY KEY(Person_Id),
    FOREIGN KEY(Person_Id) REFERENCES Person(Person_Id) ON DELETE CASCADE
);


/* Production Company Table */

CREATE TABLE Production_Company(
    Production_company_Id INTEGER NOT NULL,
    Name VARCHAR(30),
    Location VARCHAR(30),
    Address VARCHAR(200),
    PRIMARY KEY(Production_company_Id)
);
```

Producer

```
/* Produces Table */
CREATE TABLE Produces(
    Production_company_Id INTEGER NOT NULL,
    Show_Id INTEGER NOT NULL,
    FOREIGN KEY(Show_Id) REFERENCES Shows(Show_Id) ON DELETE CASCADE,
    FOREIGN KEY(Production_company_Id) REFERENCES
Production_Company(Production_company_Id) ON DELETE CASCADE,
    PRIMARY KEY(Production_company_Id,Show_Id)
);

/* Owns Table */
CREATE TABLE Owns(
    Producer_Id INTEGER NOT NULL,
    Production_company_Id INTEGER NOT NULL,
    FOREIGN KEY(Producer_Id) REFERENCES Producer(Person_Id) ON DELETE CASCADE,
    FOREIGN KEY(Production_company_Id) REFERENCES
Production_Company(Production_company_Id) ON DELETE CASCADE,
    PRIMARY KEY(Producer_Id,Production_company_Id)
);
```

Writer

```
/* Writer Table */

CREATE TABLE Writer(
    Person_Id INTEGER NOT NULL,
    PRIMARY KEY(Person_Id),
    FOREIGN KEY(Person_Id) REFERENCES Person(Person_Id) ON DELETE CASCADE
);

/* Written Table */

CREATE TABLE Written(
    Writer_Id INTEGER NOT NULL,
    Show_Id INTEGER NOT NULL,
    PRIMARY KEY(Writer_Id,Show_Id),
    FOREIGN KEY(Writer_Id) REFERENCES Writer(Person_Id) ON DELETE CASCADE,
    FOREIGN KEY(Show_Id) REFERENCES Shows(Show_Id) ON DELETE CASCADE
);
```

Cinematographer

```
/* Cinematographer Table */

CREATE TABLE Cinematographer(
    Person_Id INTEGER NOT NULL,
    PRIMARY KEY(Person_Id),
    FOREIGN KEY(Person_Id) REFERENCES Person(Person_Id) ON DELETE CASCADE
);

/* Shooting Table */

CREATE TABLE Shooting(
    Cinematographer_Id INTEGER NOT NULL,
    Show_Id INTEGER NOT NULL,
    PRIMARY KEY(Cinematographer_Id ,Show_Id),
    FOREIGN KEY(Cinematographer_Id) REFERENCES Cinematographer(Person_Id) ON DELETE
CASCADE,
    FOREIGN KEY(Show_Id) REFERENCES Shows(Show_Id) ON DELETE CASCADE
);
```

Editor

```
/* Editor Table */

CREATE TABLE Editor(
    Person_Id INTEGER NOT NULL,
    PRIMARY KEY(Person_Id),
    FOREIGN KEY(Person_Id) REFERENCES Person(Person_Id) ON DELETE CASCADE
);

/* Editing Table */

CREATE TABLE Editing(
    Editor_Id INTEGER NOT NULL,
    Show_Id INTEGER NOT NULL,
    Software_used VARCHAR(100),
    PRIMARY KEY(Editor_Id ,Show_Id),
    FOREIGN KEY(Editor_Id) REFERENCES Editor(Person_Id) ON DELETE CASCADE,
    FOREIGN KEY(Show_Id) REFERENCES Shows(Show_Id) ON DELETE CASCADE
);
```

Distributors

```
/* Distributor Table */

CREATE TABLE Distributor(
    Person_Id INTEGER NOT NULL,
    PRIMARY KEY(Person_Id),
    FOREIGN KEY(Person_Id) REFERENCES Person(Person_Id) ON DELETE CASCADE
);

/* Distributing Table*/

CREATE TABLE Distributing(
    Distributor_Id Integer NOT NULL,
    Show_Id Integer NOT NULL,
    PRIMARY KEY(Distributor_Id ,Show_Id),
    FOREIGN KEY(Distributor_Id ) REFERENCES Distributor(Person_Id) ON DELETE CASCADE,
    FOREIGN KEY(Show_Id) REFERENCES Shows(Show_Id) ON DELETE CASCADE
);
```

User tables

```
CREATE TABLE User(
    Person_Id INTEGER NOT NULL,
    User_Id VARCHAR(30),
    PRIMARY KEY(Person_Id),
    UNIQUE(User_Id),
    FOREIGN KEY(Person_Id) REFERENCES Person(Person_Id) ON DELETE CASCADE
);

CREATE TABLE Critic(
    Person_Id INTEGER NOT NULL,
    User_Id VARCHAR(30),
    PRIMARY KEY(Person_Id),
    FOREIGN KEY(Person_Id) REFERENCES User(Person_Id) ON DELETE CASCADE
);
```

User Tables

```
/* Regular User */
CREATE TABLE Regular_User(
    Person_Id INTEGER NOT NULL,
    User_Id VARCHAR(30) NOT NULL,
    PRIMARY KEY(User_Id),
    FOREIGN KEY(Person_Id) REFERENCES User(Person_Id) ON DELETE CASCADE
);

/* Reviews table */
CREATE TABLE Reviews(
    Review_Id INTEGER NOT NULL,
    User_Id VARCHAR(30) NOT NULL,
    Show_Id Integer NOT NULL,
    UP_Votes INTEGER,
    Down_Votes INTEGER,
    Rating FLOAT,
    Review_Description VARCHAR(20000),
    Reviwed_Date DATE,
    PRIMARY KEY(Review_Id),
    FOREIGN KEY(User_Id) REFERENCES User(User_Id) ON DELETE CASCADE
);
```

Movie and Collections Table

```
/* Movie Table */

CREATE TABLE Movies(
    Show_Id INTEGER NOT NULL,
    Duration FLOAT,
    Release_Date DATE,
    Year INTEGER,
    PRIMARY KEY(Show_Id),
    FOREIGN KEY(Show_Id) REFERENCES Shows(Show_Id) ON DELETE CASCADE
);

/* Box ofc collections Table */

CREATE TABLE Box_Office_Collections(
    Movie_Id INTEGER NOT NULL,
    First_Week_Collections FLOAT,
    Overall_USA_Collections FLOAT,
    Overall_Worldwide_Collections FLOAT,
    Currency VARCHAR(10),
    PRIMARY KEY(Movie_Id),
    FOREIGN KEY(Movie_Id) REFERENCES Movies(Show_Id) ON DELETE CASCADE
);
```

Genres

```
/*Genre table Schema*/
CREATE TABLE Genres(
    Genre_Id INTEGER NOT NULL,
    Name VARCHAR(30),
    PRIMARY KEY(Genre_Id)
    );

/*In_Genre table Schema*/
CREATE TABLE In_Genre(
    Genre_Id INTEGER NOT NULL,
    Show_Id INTEGER NOT NULL,
    PRIMARY KEY(Genre_Id, Show_Id),
    FOREIGN KEY(Genre_Id) REFERENCES Genres(Genre_Id),
    FOREIGN KEY(Show_Id) REFERENCES Shows(Show_Id)
 );
```

TV-Series

```
CREATE TABLE TVSeries(
    Show_Id INTEGER NOT NULL,
    Start_date DATE,
    End_date DATE,
    Air_Channel VARCHAR(50),
    Air_Day VARCHAR(30),
    Air_time TIME,
    FOREIGN KEY(Show_Id) REFERENCES Shows(Show_Id)
                    ON DELETE CASCADE,
    PRIMARY KEY(Show_Id)
);

CREATE TABLE Seasons(
    Season_Id INTEGER NOT NULL,
    Season_Name VARCHAR(30),
    Start_date DATE,
    End_date DATE,
    Season_number INTEGER,
    Show_Id Integer NOT NULL,
    FOREIGN KEY(Show_Id) REFERENCES Shows(Show_Id) ON DELETE CASCADE,
    PRIMARY KEY(Season_Id)
);
```

```
/* Episodes Table */

CREATE TABLE Episodes(
    Episode_Id Int NOT NULL,
    Episode_Title VARCHAR(50),
    Air_date DATE,
    Duration FLOAT,
    Season_Id INTEGER NOT NULL,
    FOREIGN KEY(Season_Id)
REFERENCES Seasons(Season_Id) ON
DELETE CASCADE,
    PRIMARY KEY(Episode_Id)
);
```

Award

```
/*Award Table Schema*/
CREATE TABLE Award(
    Award_Id INTEGER NOT NULL AUTO_INCREMENT,
    Name VARCHAR(50),
    Title VARCHAR(30),
    Year INTEGER,
PRIMARY KEY(Award_Id)
);
```

Persons-Nominated

```
/*Persons Nominated table Schema*/
CREATE TABLE PersonsNominated(
    Person_Id INTEGER NOT NULL,
    Award_Id INTEGER NOT NULL,
    PRIMARY KEY(Person_Id, Award_Id),
    FOREIGN KEY(Person_Id) REFERENCES Person(Person_Id),
    FOREIGN KEY(Award_Id) REFERENCES Award(Award_Id)
);
```

Shows-Nominated

```
/*Shows Nominated */
CREATE TABLE ShowsNominated(
    Show_Id INTEGER NOT NULL,
    Award_Id INTEGER NOT NULL,
    PRIMARY KEY(Show_Id, Award_Id),
    FOREIGN KEY(Show_Id) REFERENCES Shows(Show_Id),
    FOREIGN KEY(Award_Id) REFERENCES Award(Award_Id)
);
```

Presented By

```
/* Table organization */

CREATE TABLE Organization(
    Organization_Id INTEGER NOT NULL,
    Name VARCHAR(100),
    PRIMARY KEY(Organization_Id)
);
```

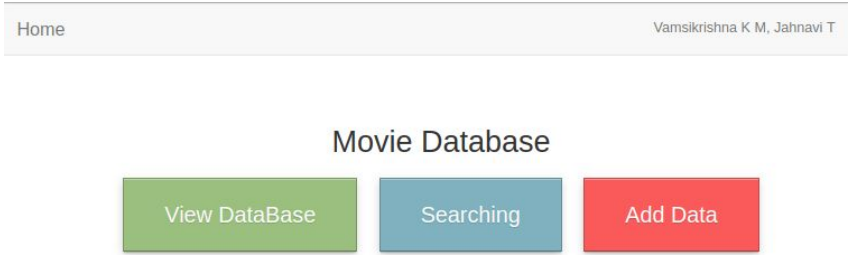
Presented By

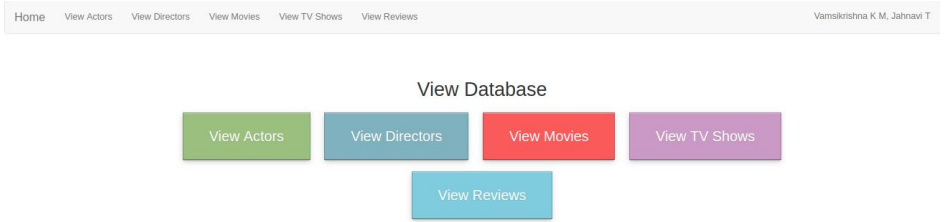
```
/*Table presented by*/
CREATE TABLE Presented_By(
    Award_Id INTEGER NOT NULL,
    Organization_Id INTEGER NOT NULL,
    PRIMARY KEY(Award_Id),
    FOREIGN KEY(Award_Id) REFERENCES Award(Award_Id),
    FOREIGN KEY(Organization_Id) REFERENCES Organization(Organization_Id)
);
```

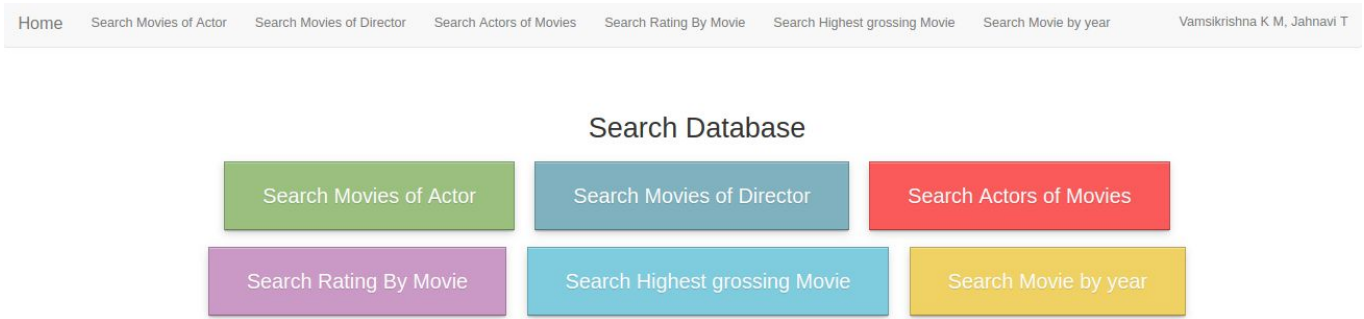
Won By

```
CREATE TABLE Won_by(
          Award_Id INTEGER NOT NULL,
          Person_Id INTEGER,
          Show_Id INTEGER,
          PRIMARY KEY(Award_Id),
          FOREIGN KEY(Award_Id) REFERENCES Awards(Award_Id),
          FOREIGN KEY(Person_Id) REFERENCES Person(Person_Id),
          FOREIGN KEY(Show_Id) REFERENCES Shows(Show_Id),
          Check ( (Show_Id IS NOT NULL AND Person_Id IS NULL) OR
                              (Show_Id IS NULL AND Person_Id IS NOT NULL)));
```

UI Snap Shots







UI Snap Shots

Add Actor

FirstName

Nick

LastName

Cassavetes

BirthDay

21/05/1959

Gender

M

Net Worth

30

Since Year

1970

Submit

Add Director

FirstName

Mark

LastName

Ruffalo

BirthDay

22/09/1967

where the

M

Direction Type

Movie

Since Year

1989

Submit

```
SELECT * FROM Director t1 JOIN Person t2 ON t1.Person_Id = t2.Person_Id;
```

Success

FirstName	LastName	DOB
John	Krasinski	1979-10-20
Anthony	Russo	1970-02-03
Patty	Jenkins	1971-07-24
Damien	Chazelle	1985-01-19
Ramin	Djawadi	1974-07-19
Matt	Duffer	1984-02-15
Ross	Duffer	1984-02-15
Chris	Evans	1981-06-13
Joe	Russo	1971-07-08
Nick	Cassavetes	1959-05-21

Contributions

- **Jahnavi** - Scraping the data from IMDB, HTML part of webpages, UX Design, Refining the table Schema (50%), SQL Commands for Viewing and Searching data.
- **Vamsikrishna** - Setting up the Environment for Database design, PHP part of Webpages, UX Design, Refining the table Schema (50%), SQL Commands for Searching and adding Data.