Information Pool System

CSE3224

Information System Design

&

Software Engineering Lab

Student Id	Name
16.02.04.007	Amreen Tarannum Alam
16.02.04.008	Mayeesha Humaira
16.02.04.014	Shimul Paul
16.02.04.024	Ashna Nawar Ahmed

Introduction:

- Information pool is an online site that will provide news and blogs on various topics, alongside ads
- Our aim today is analyse and design the ER Diagram, Relational Model, and Basic Front End of this project

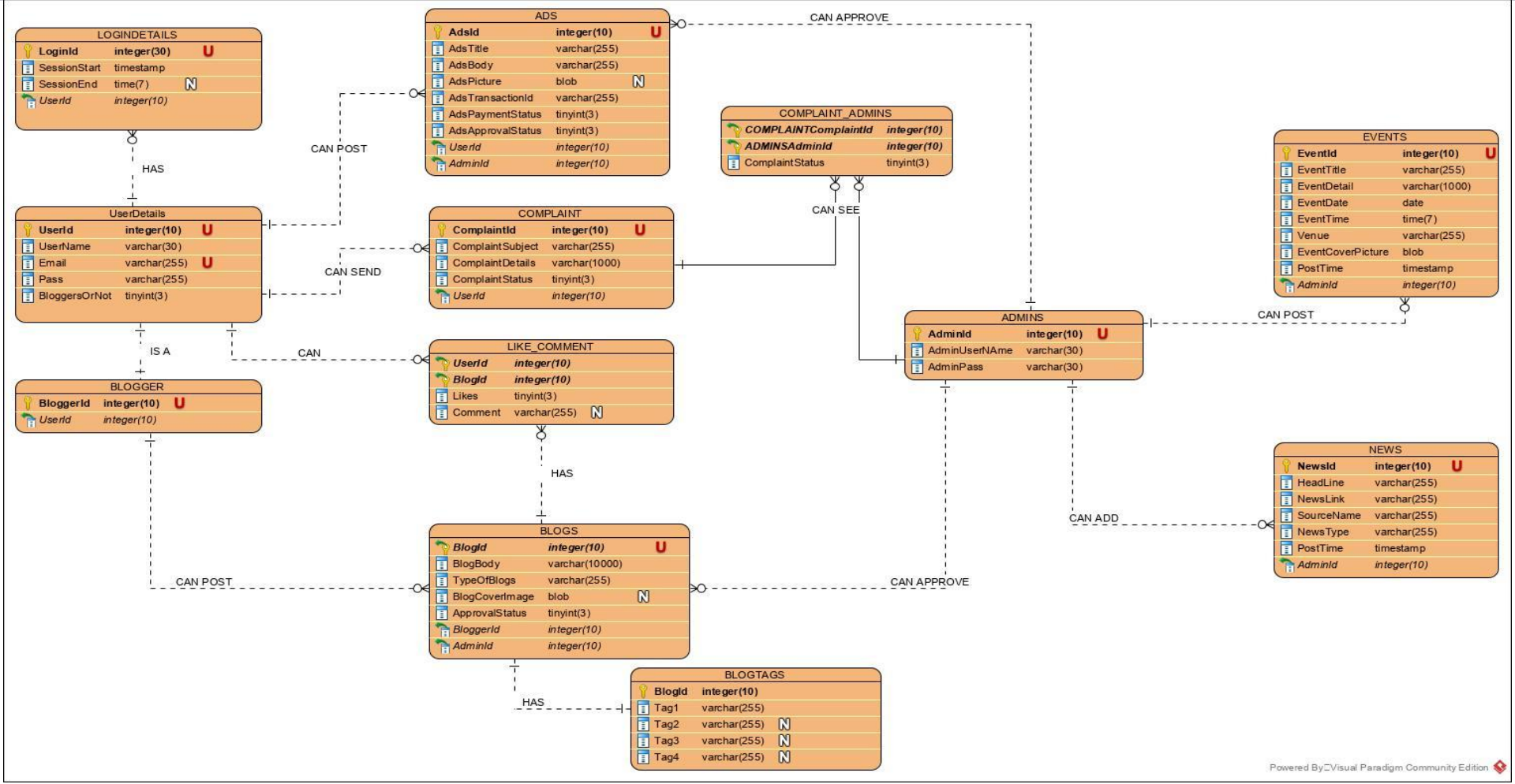
Names of Entity with Primary Key, Foreign Key or Composite Key:

- **User(UserDetails):** UserId(PK)
- **Blogger:** BloggerId(PK), *UserId(FK)*
- **Admin:** AdminId(PK)
- **LoginDetails:** LoginId(PK), *UserId(FK)*
- **Blogs:** BlogId(PK), *BloggerId(FK), AdminId(FK)*
- **BlogTags:** BlogId(PK)

Names of Entity with Primary Key, Foreign Key or Composite Key:

- **News**: NewsId(PK), *AdminId(FK)*
- **Ads**: AdsId(PK), *UserId(FK), AdminId(FK)*
- **Complaint:** ComplaintId(PK), *UserId(FK)*
- **Events**: EventId(PK), *AdminId(FK)*
- **Like_Comment**: UserId+BlogId(PK)
- **Complaint_Admins:** ComplaintId+AdminId(PK)

ER Diagram


LOGINDETAILS
LoginId integer(30) U
SessionStart timestamp
SessionEnd time(7) N
UserId integer(10)
HAS
UserDetails
UserId integer(10) U
UserName varchar(30)
Email varchar(255) U
Pass varchar(255)
BloggersOrNot tinyint(3)
IS A
BLOGGER
BloggerId integer(10) U
UserId integer(10)
CAN POST
CAN SEND
CAN
ADS
AdsId integer(10) U
AdsTitle varchar(255)
AdsBody varchar(255)
AdsPicture blob N
AdsTransactionId varchar(255)
AdsPaymentStatus tinyint(3)
AdsApprovalStatus tinyint(3)
UserId integer(10)
AdminId integer(10)
CAN APPROVE
COMPLAINT
ComplaintId integer(10) U
ComplaintSubject varchar(255)
ComplaintDetails varchar(1000)
ComplaintStatus tinyint(3)
UserId integer(10)
LIKE_COMMENT
UserId integer(10)
BlogId integer(10)
Likes tinyint(3)
Comment varchar(255) N
HAS
BLOGS
BlogId integer(10) U
BlogBody varchar(10000)
TypeOfBlogs varchar(255)
BlogCoverImage blob N
ApprovalStatus tinyint(3)
BloggerId integer(10)
AdminId integer(10)
CAN POST
CAN APPROVE
HAS
BLOGTAGS
BlogId integer(10)
Tag1 varchar(255)
Tag2 varchar(255) N
Tag3 varchar(255) N
Tag4 varchar(255) N
COMPLAINT_ADMINS
COMPLAINTComplaintId integer(10)
ADMINSAdminId integer(10)
ComplaintStatus tinyint(3)
CAN SEE
ADMINS
AdminId integer(10) U
AdminUserName varchar(30)
AdminPass varchar(30)
EVENTS
EventId integer(10) U
EventTitle varchar(255)
EventDetail varchar(1000)
EventDate date
EventTime time(7)
Venue varchar(255)
EventCoverPicture blob
PostTime timestamp
AdminId integer(10)
CAN POST
CAN ADD
NEWS
NewsId integer(10) U
HeadLine varchar(255)
NewsLink varchar(255)
SourceName varchar(255)
NewsType varchar(255)
PostTime timestamp
AdminId integer(10)
Powered By Visual Paradigm Community Edition

Relational Model

SQL Commands:

```
CREATE DATABASE ISDFINAL;
USE ISDFINAL;

/*UserDetails TABLE*/
CREATE TABLE UserDetails (
UserId INT IDENTITY(1,1) UNIQUE ,
UserName VARCHAR (30) NOT NULL,
Email VARCHAR (255) NOT NULL,
Pass VARCHAR (255) NOT NULL,
BloggerOrNot TINYINT NOT NULL DEFAULT 0,

PRIMARY KEY (USERID)
);
```

SQL Commands (Cont…):

```
/*LOGINDETAILS TABLE*/
CREATE TABLE LOGINDETAILS (
LoginId INT IDENTITY(1,1) UNIQUE,
SessionStart TIMESTAMP NOT NULL,
SessionEnd TIME NULL,
UserId INT NOT NULL,

PRIMARY KEY(LoginId),
FOREIGN KEY(UserId) REFERENCES  UserDetails (UserId)
);
```

SQL Commands (Cont…):

```
/*BLOGGER TABLE*/
CREATE TABLE BLOGGER(
BloggerId INT IDENTITY(1,1) UNIQUE,
UserId INT NOT NULL

PRIMARY KEY(BloggerId),
FOREIGN KEY (UserId) REFERENCES UserDetails(UserId)
);
```

SQL Commands:

```
/*ADMINS TABLE*/
CREATE TABLE ADMINS(
AdminId INT IDENTITY(1,1) UNIQUE,
AdminUserName VARCHAR(30) NOT NULL,
AdminPass VARCHAR(30) NOT NULL,

PRIMARY KEY (AdminId)
);
```

SQL Commands (Cont…):

```
/*BLOGS TABLE*/
CREATE TABLE BLOGS(
BlogId INT IDENTITY(1,1) UNIQUE,
BlogBody VARCHAR(1000) NOT NULL,
ApprovalStatus TINYINT NOT NULL DEFAULT 0,
TypeOfBlog VARCHAR(255) NOT NULL,
BlogCoverImage IMAGE,
BlogerId INT NOT NULL,
AdminId INT NULL,

PRIMARY KEY (BlogId),
FOREIGN KEY (BlogerId) REFERENCES Blogger(BloggerId),
FOREIGN KEY (AdminId) REFERENCES ADMINS(AdminId)
);
```

SQL Commands (Cont…):

```
/*BLOGTAGS TABLE*/
CREATE TABLE BLOGTAGS(
BlogId INT ,
Tag1 VARCHAR(255) NOT  NULL,
Tag2 VARCHAR(255) NULL,
Tag3 VARCHAR(255) NULL,
Tag4 VARCHAR(255) NULL,

PRIMARY KEY (BlogId),
FOREIGN KEY (BlogId) REFERENCES BLOGS(BlogId)
);
```

SQL Commands (Cont…):

```
/*COMPLAINT TABLE*/
CREATE TABLE COMPLAINT(
ComplaintId INT IDENTITY(1,1),
ComplaintSubject VARCHAR(255) NOT NULL,
ComplaintDetails VARCHAR(1000) NOT NULL,
ComplaintStatus TINYINT DEFAULT 0,
UserId INT NOT NULL,

PRIMARY KEY (ComplaintId),
FOREIGN KEY (UserId) REFERENCES UserDetails(UserId)
);
```

SQL Commands (Cont…):

```
/*ADS TABLE*/
CREATE TABLE ADS(
AdsId INT IDENTITY(1,1) UNIQUE,
AdsTitle VARCHAR(255) NOT NULL,
AdsBody VARCHAR(255) NOT NULL,
AdsTransactionId VARCHAR(255) NOT NULL,
AdsPaymentStatus TINYINT NOT NULL DEFAULT 0,
AdsAprovalStatus TINYINT NOT NULL DEFAULT 0,
UserId INT NOT NULL,
AdminId INT NULL,

PRIMARY KEY (AdsId),
FOREIGN KEY (UserId) REFERENCES UserDetails (UserId),
FOREIGN KEY (AdminId) REFERENCES ADMINS(AdminId)
);
```

SQL Commands (Cont…):

```
/*NEWS TABLE*/
CREATE TABLE NEWS(
NewsId INT IDENTITY(1,1) UNIQUE,
Headline VARCHAR(255) NOT NULL,
NewsLink VARCHAR(255) NOT NULL,
SourceName VARCHAR(255) NOT NULL,
NewsType VARCHAR(255) NOT NULL,
PostTime DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,
AdminId INT NOT NULL,

PRIMARY KEY(NewsId),
FOREIGN KEY (AdminId) REFERENCES ADMINS (AdminId)
);
```

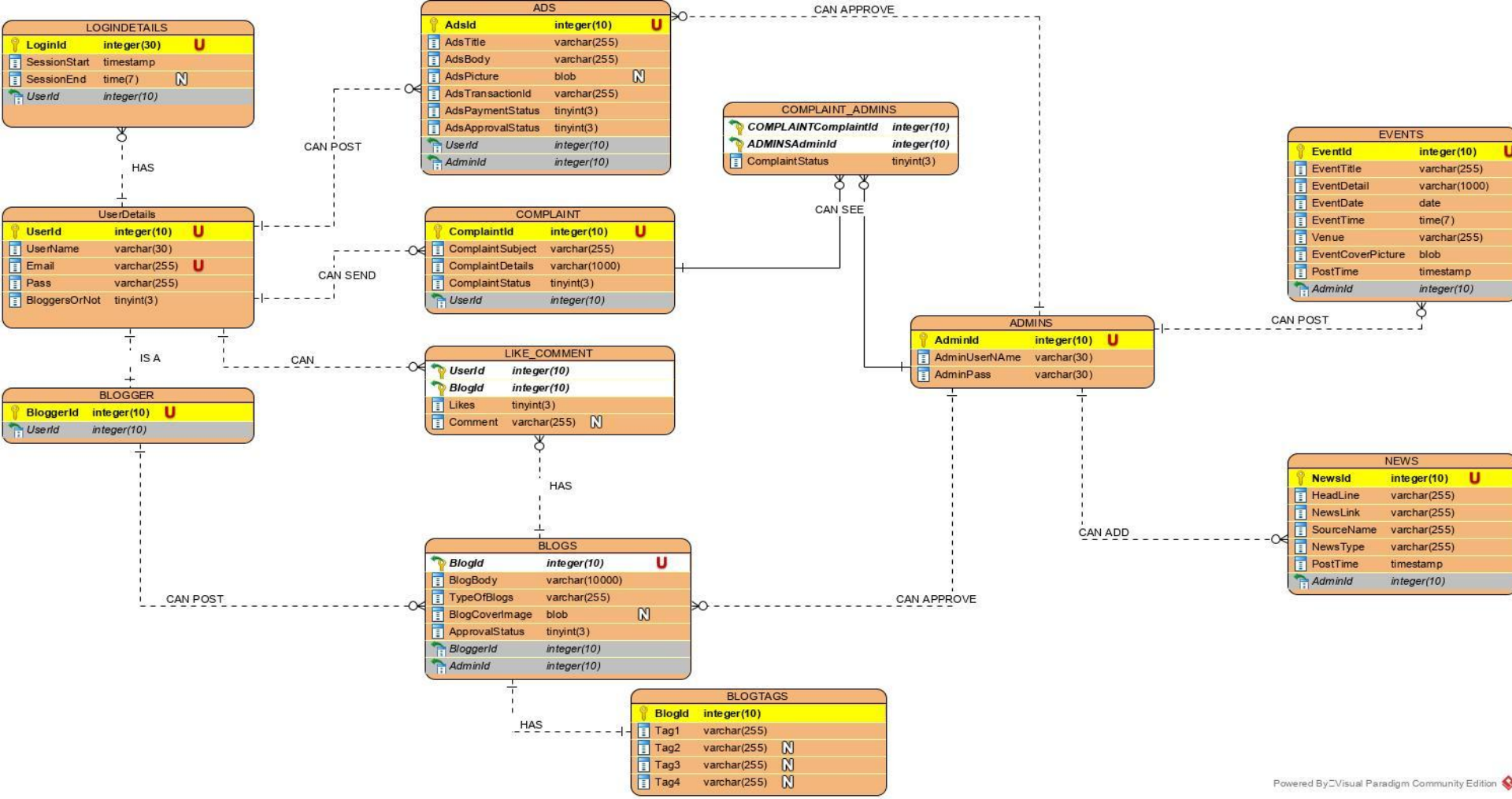
SQL Commands (Cont…):

```
/*EVENTS TABLE*/
CREATE TABLE EVENTS(
EventId INT IDENTITY(1,1) UNIQUE,
EventTitle VARCHAR(255) NOT NULL,
EventDetail VARCHAR(1000) NOT NULL,
EventDate DATE NOT NULL,
EventTime TIME NOT NULL,
Venue VARCHAR(255) NOT NULL,
EventCoverPicture IMAGE NOT NULL,
PostTime DATETIME NOT NULL
        DEFAULT CURRENT_TIMESTAMP,
AdminId INT NOT NULL,

PRIMARY KEY(EventId),
FOREIGN KEY (AdminId) REFERENCES ADMINS (AdminId)
);
```

SQL Commands (Cont…):

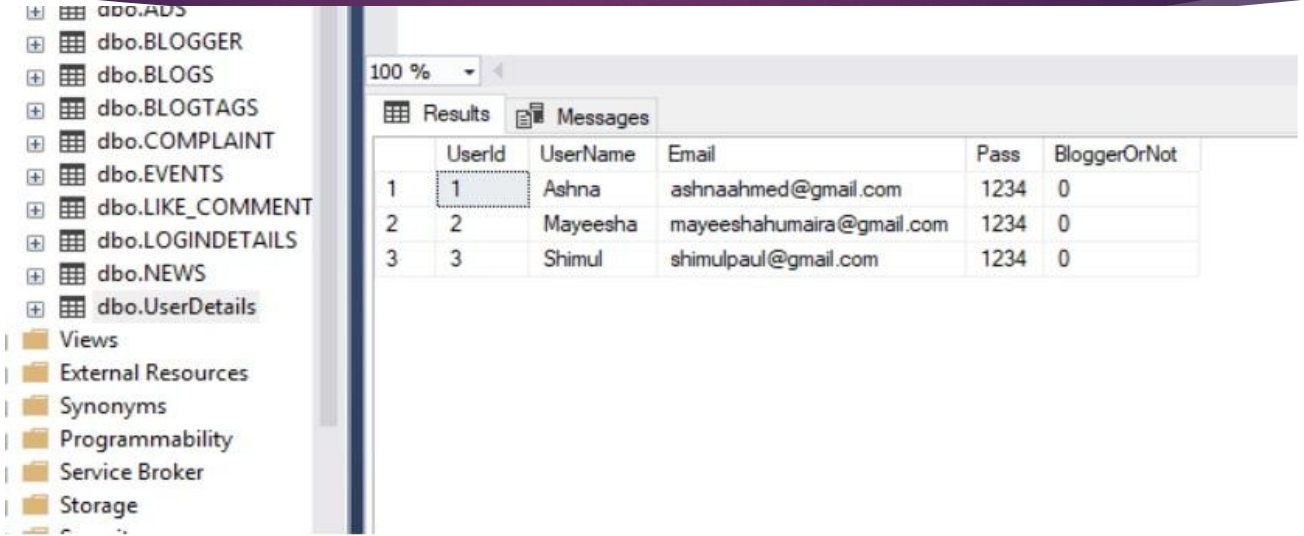
```
/*LIKE_COMMENT TABLE*/
CREATE TABLE LIKE_COMMENT(
UserId INT ,
BlogId INT ,
Likes TINYINT NOT NULL DEFAULT 0,
Comment VARCHAR(255) NULL

PRIMARY KEY(UserId,BlogId),
FOREIGN KEY (UserId) REFERENCES UserDetails(UserId),
FOREIGN KEY (BlogId) REFERENCES BLOGS(BlogId)
);
```

Highlight the Primary Key and Foreign Key

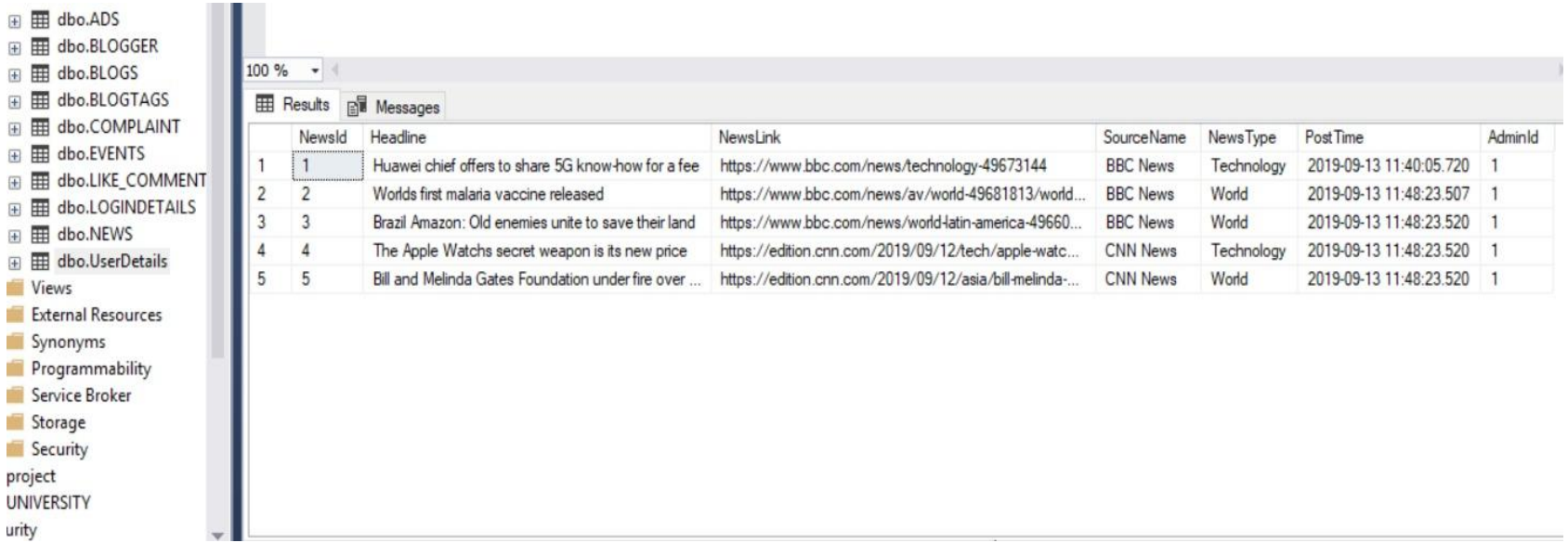

LOGINDETAILS
LoginId integer(30) U
SessionStart timestamp
SessionEnd time(7) N
UserId integer(10)
ADS
AdsId integer(10) U
AdsTitle varchar(255)
AdsBody varchar(255)
AdsPicture blob N
AdsTransactionId varchar(255)
AdsPaymentStatus tinyint(3)
AdsApprovalStatus tinyint(3)
UserId integer(10)
AdminId integer(10)
CAN APPROVE
COMPLAINT_ADMINS
COMPLAINTComplaintId integer(10)
ADMINSAdminId integer(10)
ComplaintStatus tinyint(3)
EVENTS
EventId integer(10) U
EventTitle varchar(255)
EventDetail varchar(1000)
EventDate date
EventTime time(7)
Venue varchar(255)
EventCoverPicture blob
PostTime timestamp
AdminId integer(10)
HAS
CAN POST
UserDetails
UserId integer(10) U
UserName varchar(30)
Email varchar(255) U
Pass varchar(255)
BloggersOrNot tinyint(3)
CAN SEND
COMPLAINT
ComplaintId integer(10) U
ComplaintSubject varchar(255)
ComplaintDetails varchar(1000)
ComplaintStatus tinyint(3)
UserId integer(10)
CAN SEE
ADMINS
AdminId integer(10) U
AdminUserName varchar(30)
AdminPass varchar(30)
CAN POST
IS A
CAN
LIKE_COMMENT
UserId integer(10)
BlogId integer(10)
Likes tinyint(3)
Comment varchar(255) N
BLOGGER
BloggerId integer(10) U
UserId integer(10)
NEWS
NewsId integer(10) U
HeadLine varchar(255)
NewsLink varchar(255)
SourceName varchar(255)
NewsType varchar(255)
PostTime timestamp
AdminId integer(10)
CAN ADD
HAS
BLOGS
BlogId integer(10) U
BlogBody varchar(10000)
TypeOfBlogs varchar(255)
BlogCoverImage blob N
ApprovalStatus tinyint(3)
BloggerId integer(10)
AdminId integer(10)
CAN POST
CAN APPROVE
HAS
BLOGTAGS
BlogId integer(10)
Tag1 varchar(255)
Tag2 varchar(255) N
Tag3 varchar(255) N
Tag4 varchar(255) N
Powered By Visual Paradigm Community Edition

Insert some dummy data into the Table and Justify

User Details Table:

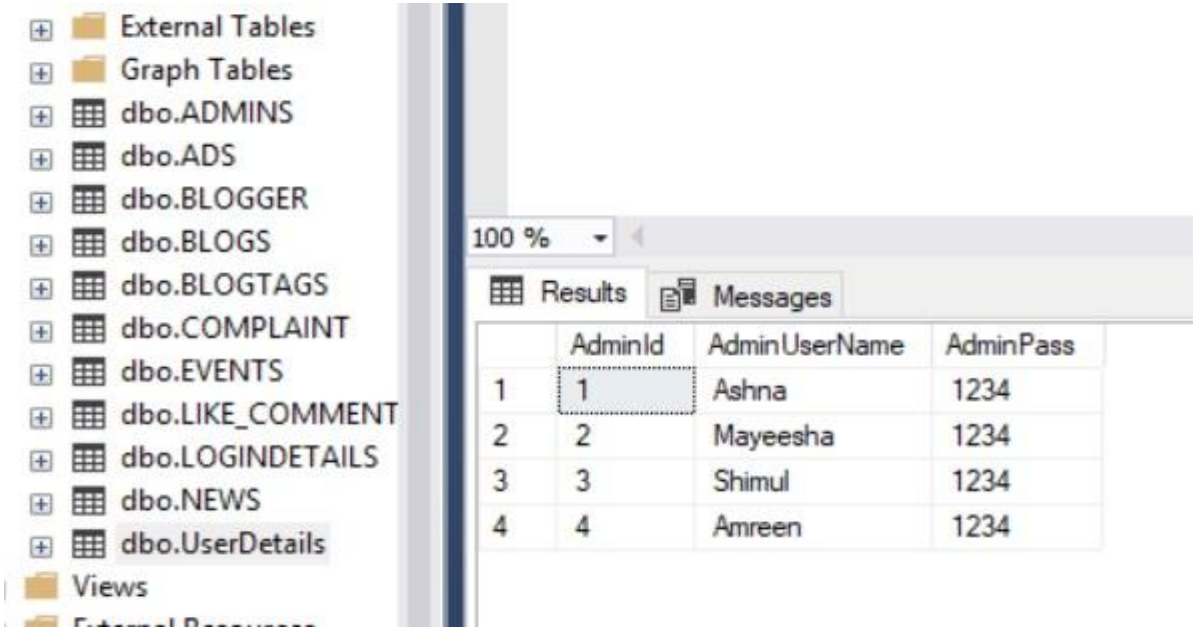


News Table:



	NewsId	Headline	NewsLink	SourceName	NewsType	PostTime	AdminId
1	1	Huawei chief offers to share 5G know-how for a fee	https://www.bbc.com/news/technology-49673144	BBC News	Technology	2019-09-13 11:40:05.720	1
2	2	Worlds first malaria vaccine released	https://www.bbc.com/news/av/world-49681813/world...	BBC News	World	2019-09-13 11:48:23.507	1
3	3	Brazil Amazon: Old enemies unite to save their land	https://www.bbc.com/news/world-latin-america-49660...	BBC News	World	2019-09-13 11:48:23.520	1
4	4	The Apple Watchs secret weapon is its new price	https://edition.cnn.com/2019/09/12/tech/apple-watc...	CNN News	Technology	2019-09-13 11:48:23.520	1
5	5	Bill and Melinda Gates Foundation under fire over ...	https://edition.cnn.com/2019/09/12/asia/bill-melinda-...	CNN News	World	2019-09-13 11:48:23.520	1

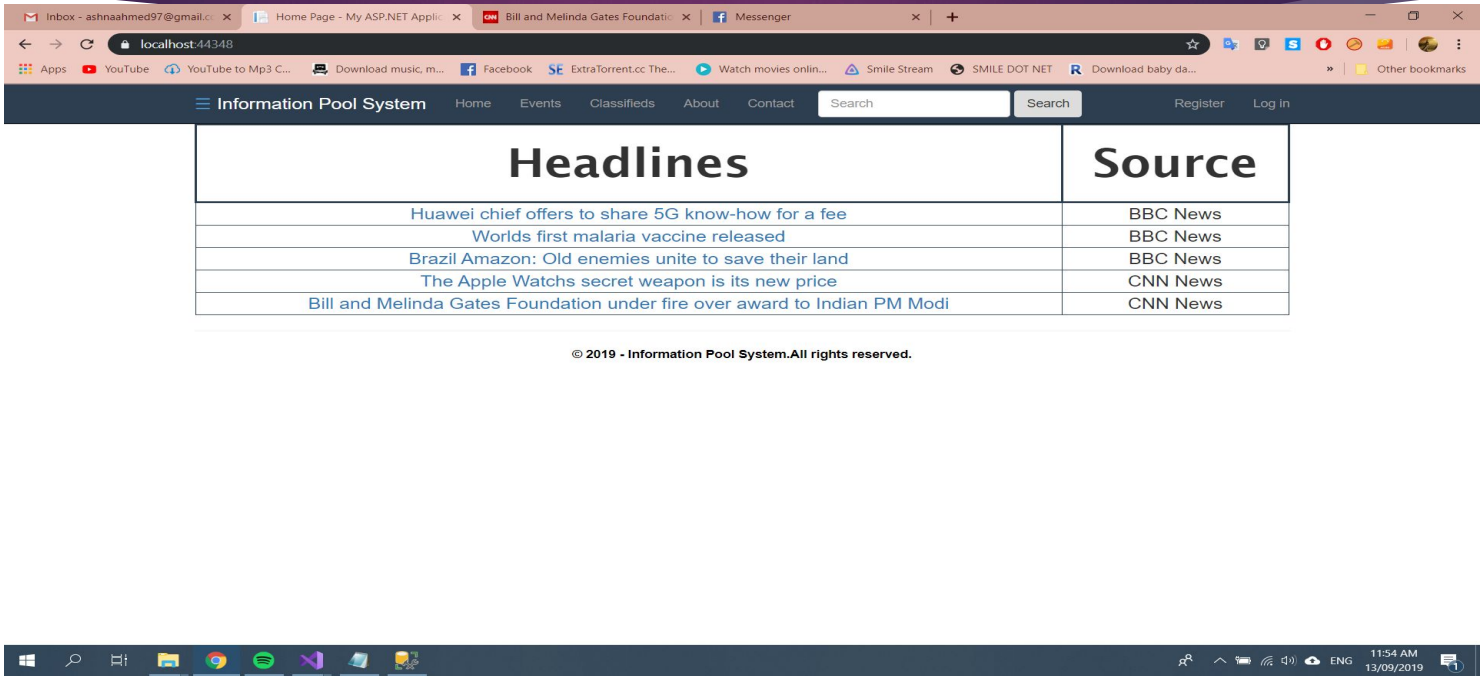
Admin Table:



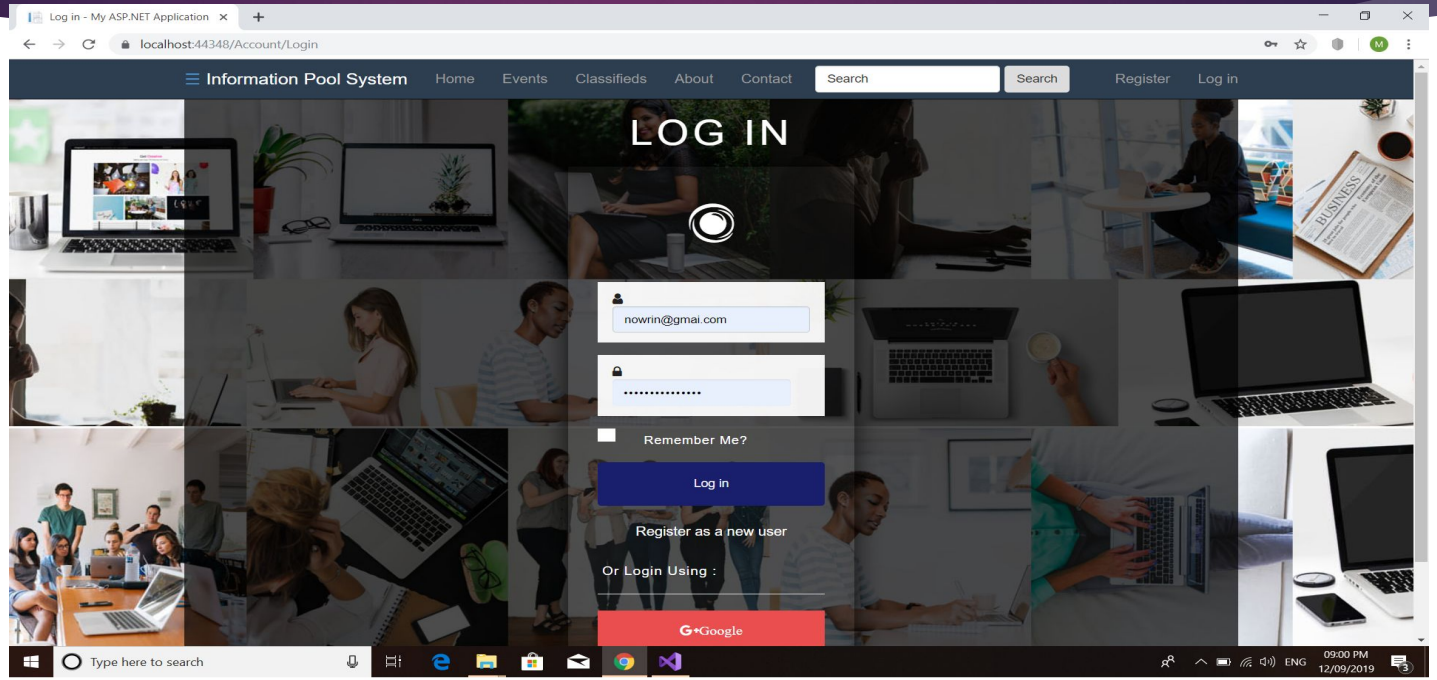
	AdminId	AdminUserName	AdminPass
1	1	Ashna	1234
2	2	Mayeesha	1234
3	3	Shimul	1234
4	4	Amreen	1234

Design:

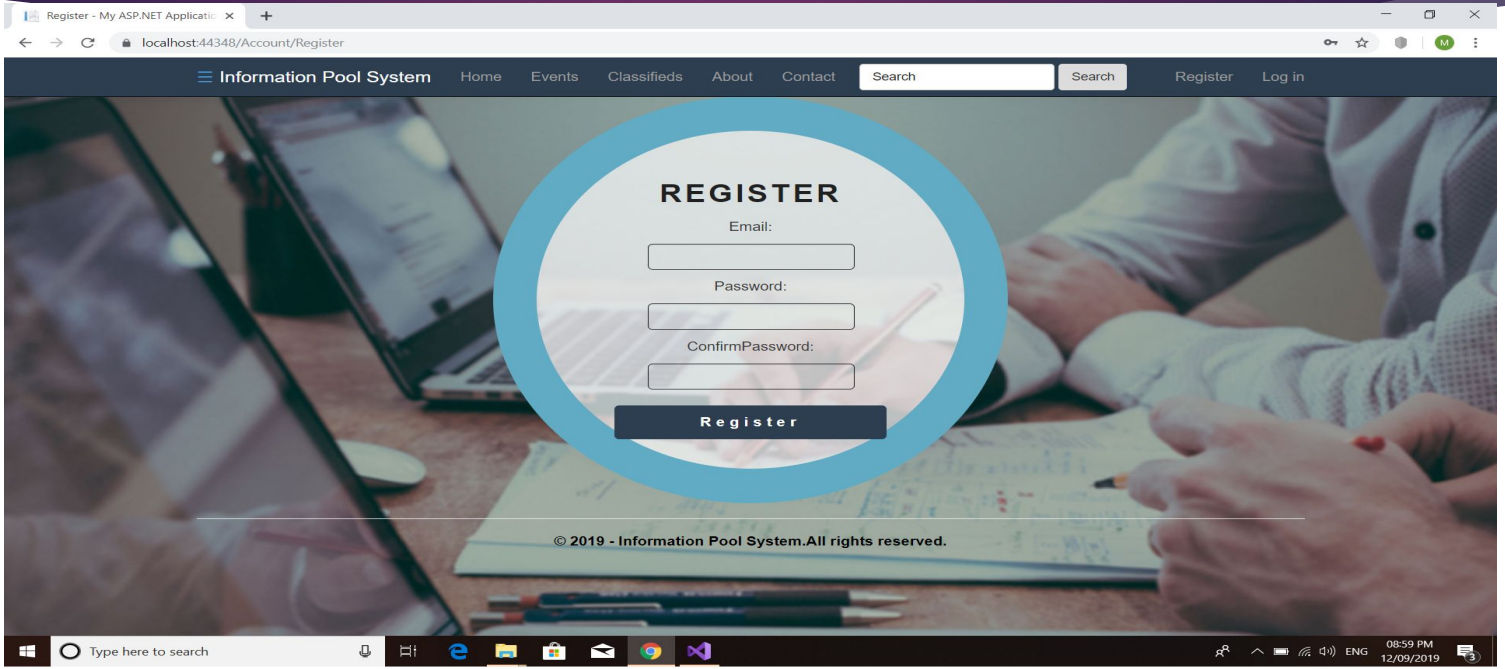
Home Page:


Inbox - ashnaahmed97@gmail.c
Home Page - My ASP.NET Applic
Bill and Melinda Gates Foundatic
Messenger
localhost:44348
Apps
YouTube
YouTube to Mp3 C...
Download music, m...
Facebook
ExtraTorrent.cc The...
Watch movies onlin...
Smile Stream
SMILE DOT NET
Download baby da...
Other bookmarks
Information Pool System
Home
Events
Classifieds
About
Contact
Search
Search
Register
Log in
Headlines
Source
Huawei chief offers to share 5G know-how for a fee
BBC News
Worlds first malaria vaccine released
BBC News
Brazil Amazon: Old enemies unite to save their land
BBC News
The Apple Watchs secret weapon is its new price
CNN News
Bill and Melinda Gates Foundation under fire over award to Indian PM Modi
CNN News
© 2019 - Information Pool System.All rights reserved.
ENG
11:54 AM
13/09/2019

Login Page:


Log in - My ASP.NET Application
localhost:44348/Account/Login
Information Pool System
Home
Events
Classifieds
About
Contact
Search
Search
Register
Log in
LOG IN
nowrin@gmai.com
Remember Me?
Log in
Register as a new user
Or Login Using :
Google
Type here to search
ENG
09:00 PM
12/09/2019

Registration Page:



Conclusion:

- We visualised the Information Pool System using ER Diagram, Relational Model, and Basic Front End.
- These diagrams will be helpful in the future to build the back end of our project