



DEPARTMENT OF COMPUTER SCIENCE

Course Name: Database Management Sys II
Course Code: CPS 562

PROJECT DOCUMENTATION

Submitted by:

**TASNEEM AMINA
SAI PRAVEENA LAKSHMI PARDESI
ANUSHA GURRALA
SHRAVANI ADYRE
SHAIK BASHEER AYESHA**

Submitted to:

Dr. BUCKLEY, JAMES

Associate Professor; Director of Graduate Program in Computer Science
University of Dayton

➤ **Introduction:-**

Facebook Mini Application is an Provides an interactive place for users, stores user profiles, post articles, upload photos search contents. A new user can register into our application by giving their information and he gets registered into the system. The registered users can login into system by giving username/password. The registered user will have a user profile and associated contents.

The Application facilitates users to search based on user profiles such as name, gender, school, groups, categories and events. The user can create an event, group and can become the owner/admin of the group. He has the facility to join the group and subscribe to particular category based on his interests or likes. Overall, the system stores information about user profiles, files, user's likes, groups and its owner/members etc.

It is a user friendly interface which uses Java Servlets, HTML, CSS and Netbeans IDE 8.2 and Glassfish Server, a complete set of development tools for building web applications. To maintain a stable database which provides high level of security and data protection Oracle Express Edition 11g is used for storing information that is entered in the graphical user interface

This mini Facebook application system has following sections

- Login Page
- Registration Page
- Welcome Page/Home Page
- Profile Page of the Registered User

1.Main features

Front End

- Registration Page for the new User
- Login Page for the Registered User
- Welcome/Home Page
- Profile Page of the User

DataBase/Back End

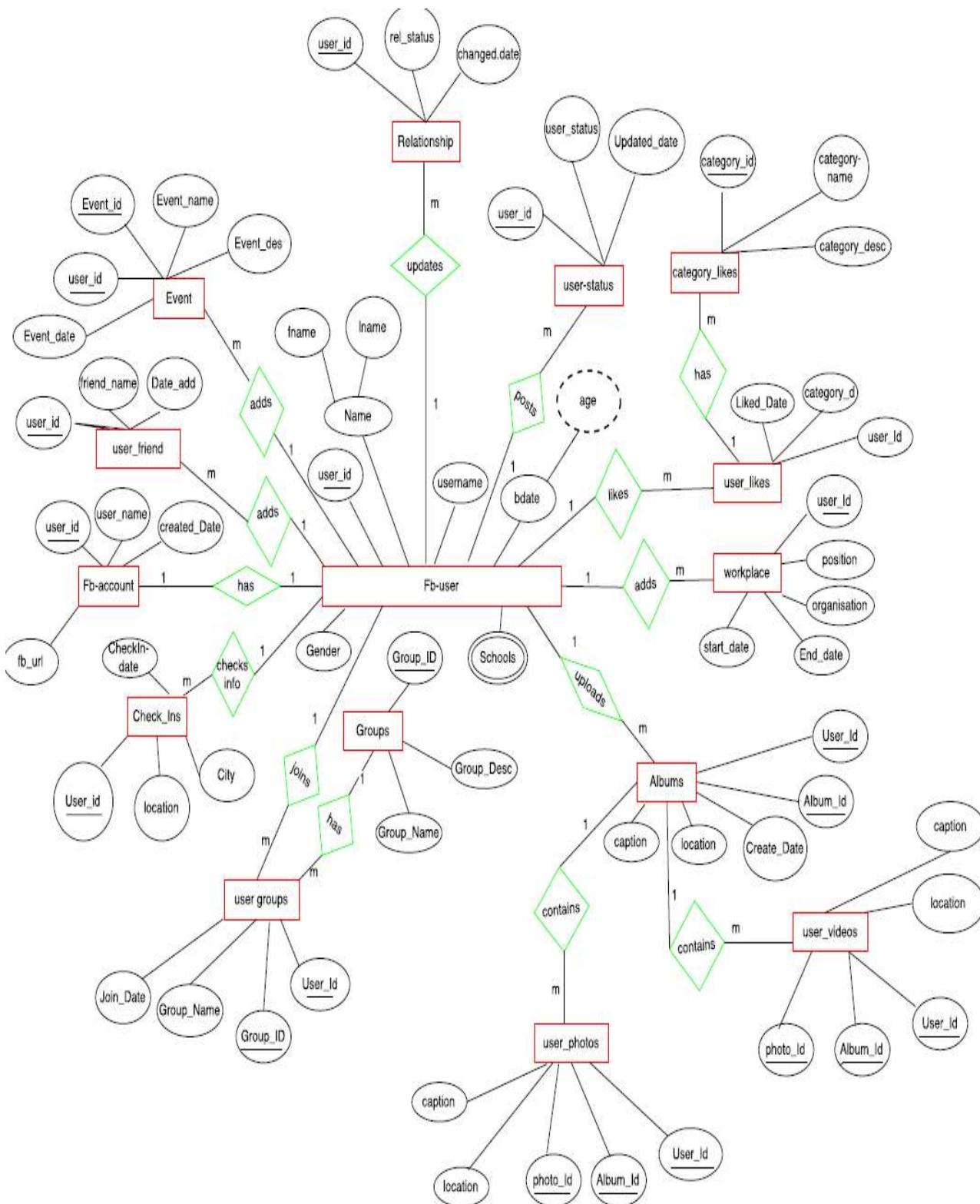
- List of Friends a User Has.
- Status of the User
- Relationship status of the user
- User Enrolled Events and Dates, Information on the Events
- His Current Activities like Category Likes
- User Likes
- Check-Ins (Places where user visits and how many times user has been to that place)
- Workplace
- Groups where user is a member.
- User Created Groups.
- Album - List of all the photos where he is been tagged.
- User Photos - Photos uploaded by user
- User Videos - Videos Uploaded by user.

2.Entity-relationship model

ER model is a data model for describing the data or information aspects of a domain or its requirements, in a way that lends itself to being implemented in a database such as relational database. The components of ER models are entities and the relationship that exists between them. Among the three levels of ER model we have developed physical data model. Physical data model is developed to be instantiated as a database. Each physical model should contain enough detail to produce a database. Physical data model can be technology independent since we are implementing database management system in a different technology. Entities are nouns and relationship demonstrates how entities are connected to one another. Entities and relationship can have attributes which of different types as follows derived attributes, multi-valued attribute, composite attribute. Every entity must have a uniquely defined attribute which is the entity's primary key. Entities are related as one-to-many or many-to-many cardinalities.

Entities

- Users
- Accounts
- User Friends
- Status
- Relationship
- Events
- Category Likes
- User Likes
- Check Ins
- Workplace
- Groups
- User Groups
- Album
- User Photos
- User Videos



Relationships:

- Users>Account(1-1)
- Users>Friends(1-M)
- Users>Status(1-M)
- Users>Relationship(1-M)
- Users>Events(M-M)
- Category_Likes>User_Likes(M-M)
- Users>Check_Ins(1-1)
- Users>Workplace(1-M)
- Users>Groups(M-M)
- Groups>User_Groups(M-M)
- Album>User_photos(M-M)
- Album>User_Videos(M-M)

Composite Attribute: Name

Derived Attribute: Age

User_Id is been the Primary Key.

3.Database Schema

Database schema is a formal language supported by the database management system. Database schema consists of integrity constraints imposed on a database, it ensures compatibility between parts of the schema.

3.1. Relational Database Schema

user_id is been used as the primary key for all the tables in the facebook mini application. whenever we register as a new user, a new unique user_id is been generated and stored in the database.

Fb_user

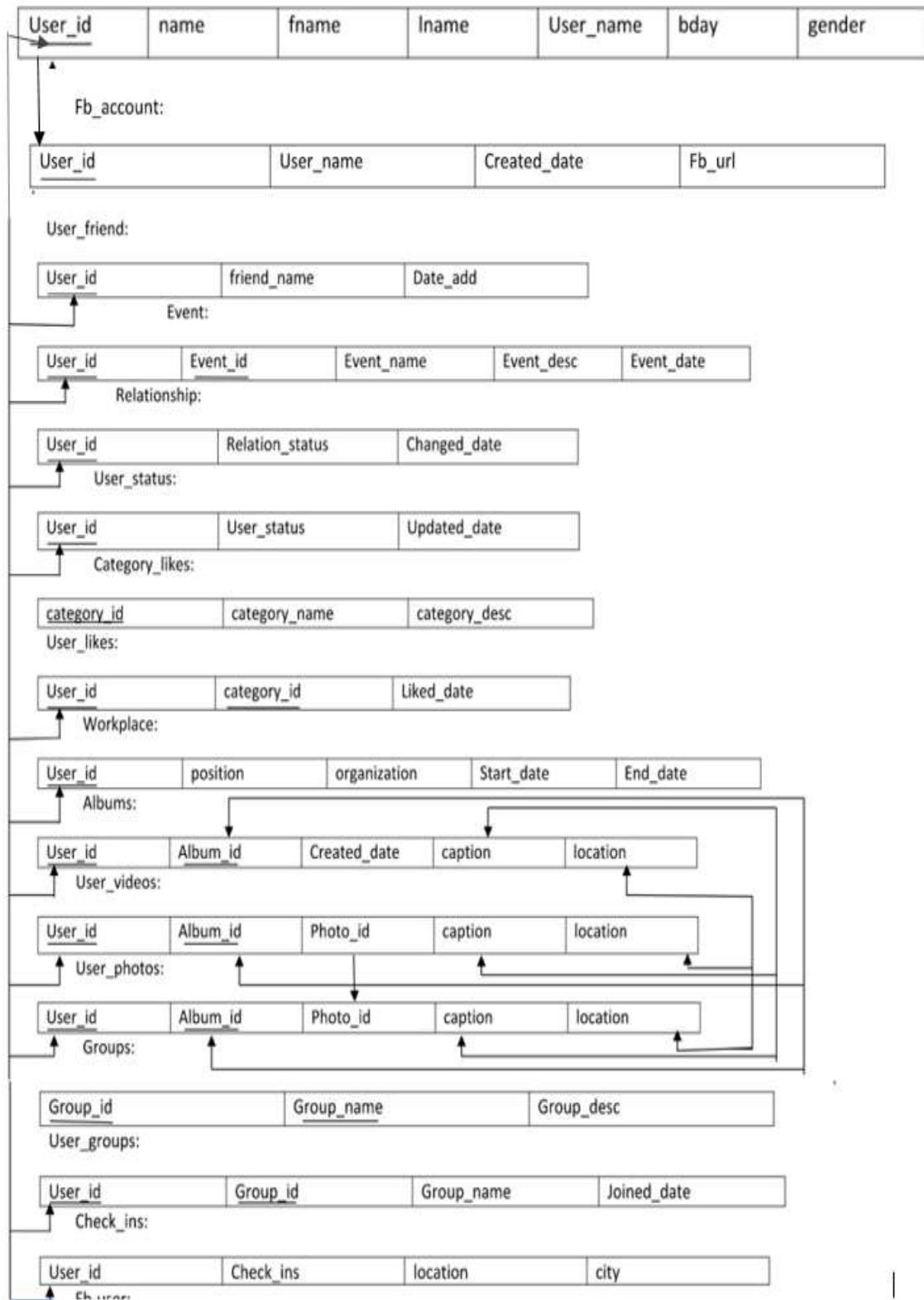


Table Name	Table Schema	Row
fb_User	User_Id number(4) not null, fname varchar2(55) not null, lname varchar2(55) not null, User_Name varchar2(55) not null, Password varchar(55) not NULL, bdate Date, Gender varchar(1), School_Name varchar2(55), primary key (User_Id));	12
fb_Account	User_Id number(4) not null, User_Name varchar2(55) not null, Created_Date DATE DEFAULT (sysdate), Fb_url varchar2(200) not null, primary key (User_id), foreign key (user_id) references fb_user(user_id));	12
User_Friend	User_Id number(4) not null, Friend_id number (4) not null, Friend_Name varchar2(55) not null, Date_Id DATE DEFAULT (sysdate), primary key (User_id,friend_id), foreign key (user_id) references fb_user(user_id));	1 friend 10 rows
User_Status	User_Id number(4) not null, User_Status varchar2(555) not null, Updated_Date DATE DEFAULT (sysdate), primary key (User_id), foreign key (user_id) references fb_user(user_id));	12
Relationship	User_Id number(4) not null, rel_status varchar2(555) not null, Changed_Date DATE DEFAULT (sysdate), primary key (User_id), foreign key (user_id) references fb_user(user_id));	12
Events	User_id number(4) not null, Event_id number(4) not null, Evet_Name varchar2(55) not null, Evet_Description varchar2(570) not null, Event_Date DATE, primary key (User_id,Event_id), foreign key (user_id) references fb_user(user_id));	1 user has 4 events

Category_Likes	Category_Id number(4) not null, Category_Name varchar2(555) not null, Category_Desc varchar2(575) not null, primary key (Category_id);	12 likes minimum
User_Likes	User_id number(4) not null, Category_Id number(4) not null, Liked_Date DATE DEFAULT (sysdate), primary key (User_id,Category_id), foreign key (user_id) references fb_user(user_id), foreign key (Category_id) references Category_Likes(Category_id));	10 - 15 likes
Check_Ins	User_id number(4) not null, Location varchar2(555) not null, City varchar2(555), CheckIn_Date DATE DEFAULT (sysdate), primary key (User_Id), foreign key (user_Id) references fb_user(user_Id));	3 - 5 check-ins per user
WorkPlace	User_Id number(4) not null, Organization varchar2(555), Position varchar2(555), Start_Date DATE not null, End_Date DATE, primary key (User_Id), foreign key (user_Id) references fb_user(user_Id));	12
Groups	Group_Id number(4) not null, Group_Name varchar2(120), Group_Type varchar2(120), primary key (Group_Id));	1 - 5 per user
User_Groups	User_Id number(4) not null, Group_Id number(4) not null, Group_Name varchar2(20), Join_Date DATE not null, primary key (User_Id,Group_Id), foreign key (user_Id) references fb_user(user_Id), foreign key (Group_Id) references Groups(Group_Id));	1- 5 per user
Album	User_Id number(4) not null, Album_Id number(4) not null, Create_Date DATE DEFAULT (sysdate),	4 - 5 albums per user.

	Location varchar2(55), Caption varchar2(50), primary key (User_Id,Album_Id), foreign key (user_Id) references fb_user(user_Id));	
User_Photos	User_Id number(4) not null, Album_Id number(4) not null, Photo_Id number(4) not null, Location varchar2(55), Caption varchar2(50), primary key (User_Id,Album_Id,Photo_Id), foreign key (User_Id,Album_Id) references Album(User_Id,Album_Id) ();	1 user has 10 -15 photos
User_Videos	User_Id number(4) not null, Album_Id number(4) not null, Video_Id number(4) not null, Location varchar2(55), Caption varchar2(50), primary key (User_Id,Album_Id,Video_Id), foreign key (User_Id,Album_Id) references Album(User_Id,Album_Id) ();	1 user has 10 -15 videos

4.Database Instance

A database state at a specific time defined through the defined through the currently existing content ,relationship and their attributes.

4.1 fb_user Instance : Below Screenshot displays all the user information such as user_id, user first name and user last name, registered username, user generated password, user date of birth, user gender and user school name.

ORACLE® Application Express

Welcome TASNEEM ([Logout](#))

Home Application Builder SQL Workshop Team Development Administration

Home > SQL Workshop > SQL Commands

Schema: TASNEEM

Autocommit Rows 10 Save Run

```
select * from fb_user;
```

Results Explain Describe Saved SQL History

USER_ID	FNAME	LNAME	USER_NAME	PASSWORD	BDATE	GENDER	SCHOOL_NAME
101	Emma	Stone	Emma101	password	01/01/1991	F	HOLLYWOOD
102	Johnny	Depp	Johnny102	password	02/02/1991	M	HOLLYWOOD
103	Al	Pacino	Al103	password	03/03/1991	M	HOLLYWOOD
104	Kevin	Spacy	Kevin104	password	04/04/1991	M	HOLLYWOOD
105	Brad	Pitt	Brad105	password	05/05/1991	M	HOLLYWOOD
106	Angelina	Jolie	Angelina106	password	06/06/1991	F	HOLLYWOOD
107	Leonardo	DiCaprio	Leonardo107	password	07/07/1991	M	HOLLYWOOD
108	Tom	Cruise	Tom108	password	08/08/1991	M	HOLLYWOOD
109	Arnold	Schwarzenegger	Arnold109	password	09/09/1991	M	HOLLYWOOD
110	Sylvester	Stallone	Sylvester110	password	10/10/1991	M	HOLLYWOOD

More than 10 rows available. Increase rows selector to view more rows.

10 rows returned in 0.00 seconds [Download](#)

4.2 fb_Account: Displays user id, user name, Fb_url and created date

E Application Express

Welcome TAS

Application Builder SQL Workshop Team Development Administration

Workshop > SQL Commands

Schema: TASNEEM

Rows 10 Save Run

```
fb_Account;
```

Results Explain Describe Saved SQL History

ER_NAME	CREATED_DATE	FB_URL
Emma101	04/24/2017	https://www.facebook.com/pages/Emma-Stone/106183659413100?ref=br_rs
Johnny102	04/24/2017	https://www.facebook.com/pages/Johnny-Depp/106076598424196
Al103	04/24/2017	https://www.facebook.com/AlPacino/?fref=105685019464762
Kevin104	04/24/2017	https://www.facebook.com/KevinSpacey/
Brad105	04/24/2017	https://www.facebook.com/pages/Brad-Pitt/105960549434629
Angelina106	04/24/2017	https://www.facebook.com/pages/Angelina-Jolie/108395679184728
Leonardo107	04/24/2017	https://www.facebook.com/LeonardoDiCaprio/
Tom108	04/24/2017	https://www.facebook.com/officialtmcruse/
Arnold109	04/24/2017	https://www.facebook.com/arnold/
Sylvester110	04/24/2017	https://www.facebook.com/SylvesterStallone/

More than 10 rows available. Increase rows selector to view more rows.

10 rows returned in 0.00 seconds [Download](#)

4.3 User_Friend: Displays User Id, Friend Id, Friend Name and Date Created

ORACLE Application Express

Welcome TASNEEM ([Logout](#))

Home Application Builder SQL Workshop Team Development Administration

Home > SQL Workshop > SQL Commands

Schema: TASNEEM

Autocommit Rows: 10 Save Run

```
select * from User_Friend;
```

Results Explain Describe Saved SQL History

USER_ID	FRIEND_ID	FRIEND_NAME	DATE_ID
101	102	Johnny102	04/24/2017
101	103	Al103	04/24/2017
101	104	Kevin104	04/24/2017
102	101	Emma101	04/24/2017
102	103	Al103	04/24/2017
102	104	Kevin104	04/24/2017
103	101	Johnny102	04/24/2017
103	102	Al103	04/24/2017
103	104	Kevin104	04/24/2017
104	101	Emma101	04/24/2017

More than 10 rows available. Increase rows selector to view more rows.

10 rows returned in 0.01 seconds [Download](#)

4.4 User_Status: User Id, User status and Date

Home Application Builder SQL Workshop Team Development Administration

Home > SQL Workshop > SQL Commands

Schema: TASNEEM

Autocommit Rows: 10 Save Run

```
select * from User_Status;
```

Results Explain Describe Saved SQL History

USER_ID	USER_STATUS	UPDATED_DATE
101	Celebrating 25th Bday	04/24/2017
102	Watching Transformers	04/24/2017
103	Thinking about DMBS Project	04/24/2017
104	Feeling happy	04/24/2017
105	Feeling sad	04/24/2017
106	Feeling emotional	04/24/2017
107	Feeling devotional	04/24/2017
108	Feeling blessed	04/24/2017
109	Feeling excited	04/24/2017
110	Feeling annoying	04/24/2017

More than 10 rows available. Increase rows selector to view more rows.

4.5 Relationship:

Displays the relationship status of the user with updated date

ORACLE® Application Express

Welcome TASNEEM (Logout)

Home Application Builder SQL Workshop Team Development Administration

Schema TASNEEM

select * from Relationship

Autocommit Rows 10 Save Run

Results Explain Describe Saved SQL History

USER_ID	REL_STATUS	CHANGED_DATE
101	Single	04/24/2017
102	Married	04/24/2017
103	Committed	04/24/2017
104	Single	04/24/2017
105	Married	04/24/2017
106	Committed	04/24/2017
107	Single	04/24/2017
108	Married	04/24/2017
109	Committed	04/24/2017
110	Single	04/24/2017

10 rows returned in 0.00 seconds Download

Application Express 4.0.2.0.09

4.6 Events:

Displays user and event Id then event description and event date

ORACLE® Application Express

Welcome TASNEEM (Logout)

Home Application Builder SQL Workshop Team Development Administration

Schema TASNEEM

select * from Events

Autocommit Rows 10 Save Run

Results Explain Describe Saved SQL History

USER_ID	EVENT_ID	EVEV_NAME	EVEV_DESCRIPTION	EVENT_DATE
101	301	A Taste of International flavors	We are always exploring and working tirelessly to discover flavors fragrances and cosmetic ingredients that inspire our customers and delight consumers	01/01/2017
102	302	Arabian Nights	A famous collection of Persian, Indian, and Arabian folktales.	01/11/2017
103	303	Caribbean Nights	Caribbean Nights - A Valentine Night at the Museum	02/10/2017
104	304	Hawaiian Paradise	Paradise Park	03/11/2017
105	305	Magic Under the Stars	The 2nd Annual Magic Under the Stars Concert featuring The Original Lakeside, Confunkshun, and The Dazz Band. This awesome lineup will perform Friday	04/01/2017
106	306	Underwater Paradise	The Unexpected journey of Mr. B in the colourful underwater world of Bubblelandia. The Underwater Paradise is being performed at Volkswagen Arena	01/12/2017
107	307	Space Odyssey	The Space Odyssey series is a series of science fiction novels by the writer Arthur C. Clarke.	02/10/2017
108	308	Shanghai Moon	Your invited to attend our 2015 museum night out: Under the Shanghai Moon Location will be at Butte Creek Country	09/09/2017
109	309	Romeo and Juliet	You are Invited to attend our Romeo and Juliet Show Tonight in NewYork	07/07/2017
110	310	One Upon a Time	Love Bella Creations presents Once Upon A Time Bridal Show Hosted By The ... Spring Open House Vendor Event 4/22 tickets.	08/08/2017

More than 10 rows available. Increase rows selector to view more rows.

10 rows returned in 0.01 seconds Download

127.0.0.1:8080/apex/f?p=4500:1000:2934980690220545

4.7 Category_Likes: Displays category id, name and the category description

ORACLE® Application Express

Welcome TASNEEM (Logout)

Home Application Builder SQL Workshop Team Development Administration

Home > SQL Workshop > SQL Commands Schema TASNEEM

Autocommit Rows 10 Save Run

```
select * from Category_Likes
```

Results Explain Describe Saved SQL History

CATEGORY_ID	CATEGORY_NAME	CATEGORY_DESC
410	j k rowling	Author
411	Roger Federer	Sports
402	Selena Gomez	Music
401	Justin Bieber	Music
403	Brad Pitt	Actor
404	Vin Diesel	Actor
405	Angelina Jolie	Actress
406	Megan Fox	Actress
407	Ellen Morgan	Social
408	Barack Obama	Social

More than 10 rows available. Increase rows selector to view more rows.

10 rows returned in 0.00 seconds Download

127.0.0.1:8080/apex/f?p=4500:1000:2934980690220545

4.8 User_Likes: Displays User id and the category id that a user has liked and the date

ORACLE® Application Express

Welcome TASNEEM (Logout)

Home Application Builder SQL Workshop Team Development Administration

Home > SQL Workshop > SQL Commands Schema TASNEEM

Autocommit Rows 10 Save Run

```
select * from User_Likes
```

Results Explain Describe Saved SQL History

USER_ID	CATEGORY_ID	LIKED_DATE
101	401	04/24/2017
101	402	04/24/2017
101	403	04/24/2017
102	402	04/24/2017
103	403	04/24/2017
104	404	04/24/2017
105	405	04/24/2017
106	406	04/24/2017
107	407	04/24/2017
108	408	04/24/2017

More than 10 rows available. Increase rows selector to view more rows.

10 rows returned in 0.00 seconds Download

4.9 Check_Ins: Displays the check_in locations, city and date

The screenshot shows the Oracle Application Express interface. The top navigation bar includes links for Home, Application Builder, SQL Workshop, Team Development, Administration, and a Schema dropdown set to TASNEEM. The SQL Workshop tab is active. The main area contains a SQL command window with the following content:

```
select * from Check_Ins
```

Below the command window is a results grid:

USER_ID	LOCATION	CITY	CHECKIN_DATE
101	Central Park	New York	02/03/2000
102	Empire State Building	New York	03/03/2001
103	Statue of Liberty	New York	04/04/2002
104	Metropolitan Museum of Art	New York	05/05/2003
112	Little Italy	New York	12/12/2011
111	Chinatown	New York	11/11/2010

Below the grid, it says "6 rows returned in 0.00 seconds" and has a "Download" link. The bottom status bar indicates "Workspace: TASNEEM User: TASNEEM" and "Language: en | Copyright © 1999, 2010, Oracle. All rights reserved.".

4.10 Workplace: Displays user id, name of the organization, user position start and end date.

The screenshot shows the Oracle Application Express interface. The top navigation bar includes links for Home, Application Builder, SQL Workshop, Team Development, Administration, and a Schema dropdown set to TASNEEM. The SQL Workshop tab is active. The main area contains a SQL command window with the following content:

```
select * from Workplace
```

Below the command window is a results grid:

USER_ID	ORGANIZATION	POSITION	START_DATE	END_DATE
101	Amazon	Software Developer	01/01/2015	-
102	Google	Business Analyst	01/02/2014	10/02/2016
103	LCS	Tester	01/03/2001	-
104	Pfizer	Director	01/03/2001	-
105	Infosys	Architect	12/04/2002	12/04/2015
106	International Movies	Writer	01/05/2000	10/05/2010
107	Hollywood Movies	Assistant Editor	12/05/2003	03/05/2003
108	University of Dayton	Assistant Professor	01/06/2003	-
109	Google	Designer	01/07/2004	10/07/2016
110	Hollywood Movies	Makeup Artist	06/07/2004	06/07/2004

Below the grid, it says "More than 10 rows available. Increase rows selector to view more rows." and "10 rows returned in 0.01 seconds". There is a "Download" link. The bottom status bar indicates "Application Express 4.0.2.00.09" and "Language: en | Copyright © 1999, 2010, Oracle. All rights reserved.".

4.11 GROUPS: Displays Group Id Group name and Group Type which is the Group category

ORACLE® Application Express

Welcome TASNEEM (Logout)

Home Application Builder SQL Workshop Team Development Administration

Schema: TASNEEM

select * from GROUPS |

Autocommit Rows 10 Save Run

Results Explain Describe Saved SQL History

GROUP_ID	GROUP_NAME	GROUP_TYPE
608	The Queen Bees	Ladies Groups
609	Friends Forever	Friends
610	Best Family	Family
611	Wandering Minds	Friends
601	Happy House	Family
602	Graveyord	Friends
603	Bonding	Siblings
612	World of Cousins	Cousins
604	Mad House	Family
605	Rocksters	Friends

More than 10 rows available. Increase rows selector to view more rows.

10 rows returned in 0.00 seconds Download

4.12 User_Groups: Displays User id along with the Group id and Group name and join date

① 127.0.0.1:8080/apex/f?p=4500:1003:2934980690220545::NO::

ORACLE® Application Express

Welcome TASNEEM (Logout)

Home Application Builder SQL Workshop Team Development Administration

Schema: TASNEEM

select * from User_Groups |

Autocommit Rows 10 Save Run

Results Explain Describe Saved SQL History

USER_ID	GROUP_ID	GROUP_NAME	JOIN_DATE
104	604	Family	04/01/2000
105	605	Friends	05/01/2000
106	606	Family	06/01/2000
107	607	Cousins	07/01/2000
108	608	Ladies Groups	08/01/2000
109	609	Friends	09/01/2000
110	610	Family	10/01/2000
111	611	Friends	11/01/2000
112	612	Cousins	12/01/2000
101	601	Family	01/01/2000

More than 10 rows available. Increase rows selector to view more rows.

10 rows returned in 0.01 seconds Download

4.13 Album: Displays user id, album id, its creation date location with caption or tag line

← → C | ① 127.0.0.1:8080/apex/f?p=4500:1003:2934980690220545::NO:::

ORACLE® Application Express

Welcome TASNEEM (Logout)

Home Application Builder SQL Workshop Team Development Administration

Home > SQL Workshop > SQL Commands Schema TASNEEM Help

Autocommit Rows 10 Save Run

```
select * from Album
```

Results Explain Describe Saved SQL History

USER_ID	ALBUM_ID	CREATE_DATE	LOCATION	CAPTION
102	702	04/25/2017	Cincinnati	International Fest
101	701	04/25/2017	Dayton	Diya
103	703	04/25/2017	Kentucky	Dandiya
104	704	04/25/2017	Columbus	Holi
105	705	04/25/2017	Chicago	Colours
112	712	04/25/2017	Alabama	Chakra
111	711	04/25/2017	North Carolina	Ank
110	710	04/25/2017	Atlanta	CodeFest
109	709	04/25/2017	Michigan	Citric
108	708	04/25/2017	New Jersey	Concetto

More than 10 rows available. Increase rows selector to view more rows.

10 rows returned in 0.01 seconds Download

4.14 User_Photos:Displays user id, album id of the photo and photo id, its location and caption

← → C | ① 127.0.0.1:8080/apex/f?p=4500:1003:2934980690220545::NO:::

ORACLE® Application Express

Welcome TASNEEM (Logout)

Home Application Builder SQL Workshop Team Development Administration

Home > SQL Workshop > SQL Commands Schema TASNEEM Help

Autocommit Rows 10 Save Run

```
select * from User_Photos
```

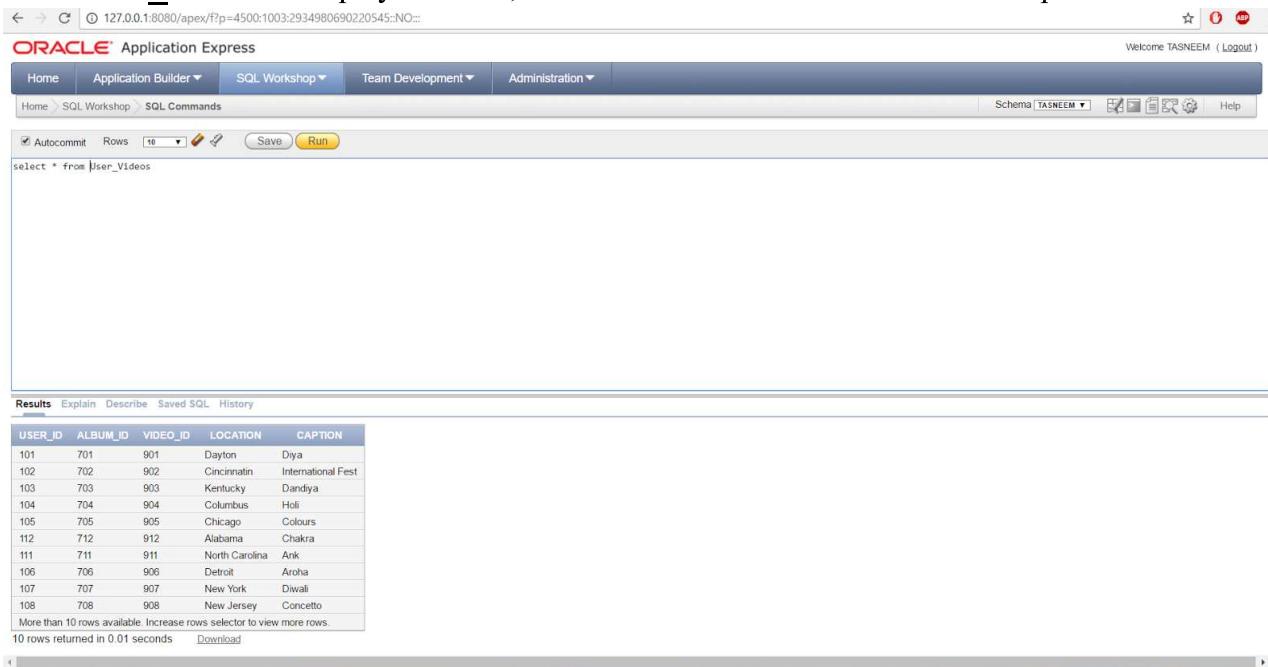
Results Explain Describe Saved SQL History

USER_ID	ALBUM_ID	PHOTO_ID	LOCATION	CAPTION
101	701	801	Dayton	Diya
102	702	802	Cincinnati	International Fest
103	703	803	Kentucky	Dandiya
104	704	804	Columbus	Holi
105	705	805	Chicago	Colours
106	706	806	Detroit	Aroha
107	707	807	New York	Diwali
108	708	808	New Jersey	Concetto
109	709	809	Michigan	Citric
110	710	810	Atlanta	CodeFest

More than 10 rows available. Increase rows selector to view more rows.

10 rows returned in 0.00 seconds Download

4.15 User_Videos: Displays User Id, album and video Id with location and caption



The screenshot shows the Oracle Application Express SQL Workshop interface. The URL is 127.0.0.1:8080/apex/f?p=4500:1003:2934980690220545::NO:::. The title bar says "ORACLE Application Express". The top menu includes Home, Application Builder, SQL Workshop, Team Development, Administration, and a welcome message for "TASNEEM" with a logout link. The SQL Workshop menu bar has "Home", "SQL Workshop", and "SQL Commands". The toolbar includes "Save" and "Run" buttons. The SQL editor contains the query: "select * from User_Videos". The results section displays a table with the following data:

USER_ID	ALBUM_ID	VIDEO_ID	LOCATION	CAPTION
101	701	901	Dayton	Diya
102	702	902	Cincinnati	International Fest
103	703	903	Kentucky	Dandya
104	704	904	Columbus	Holi
105	705	905	Chicago	Colours
112	712	912	Alabama	Chakra
111	711	911	North Carolina	Ank
106	706	906	Detroit	Aroha
107	707	907	New York	Diwall
108	708	908	New Jersey	Concerto

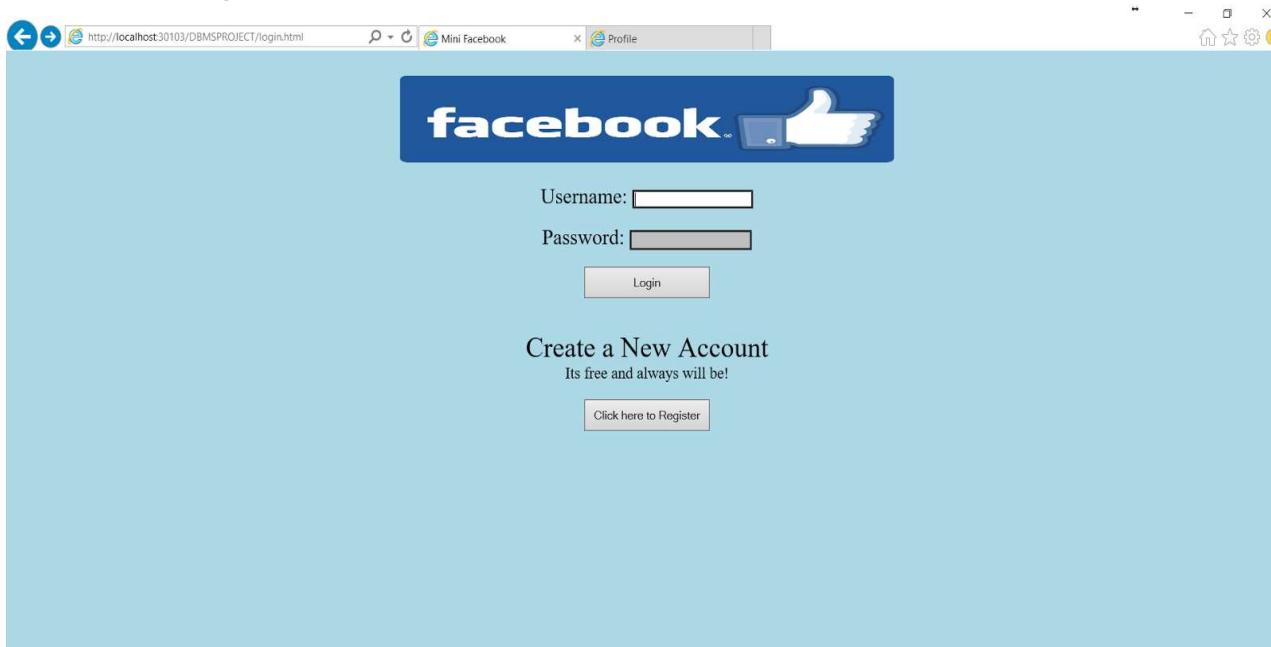
More than 10 rows available. Increase rows selector to view more rows.
10 rows returned in 0.01 seconds [Download](#)

5. Data Manipulation

Data Manipulation Language is used for selecting, inserting, deleting and updating data in a database. Read-only queries of data is considered DML. Structured Query Language is used to retrieve and manipulate data in relational database.

Web Interface

5.1: Home Page

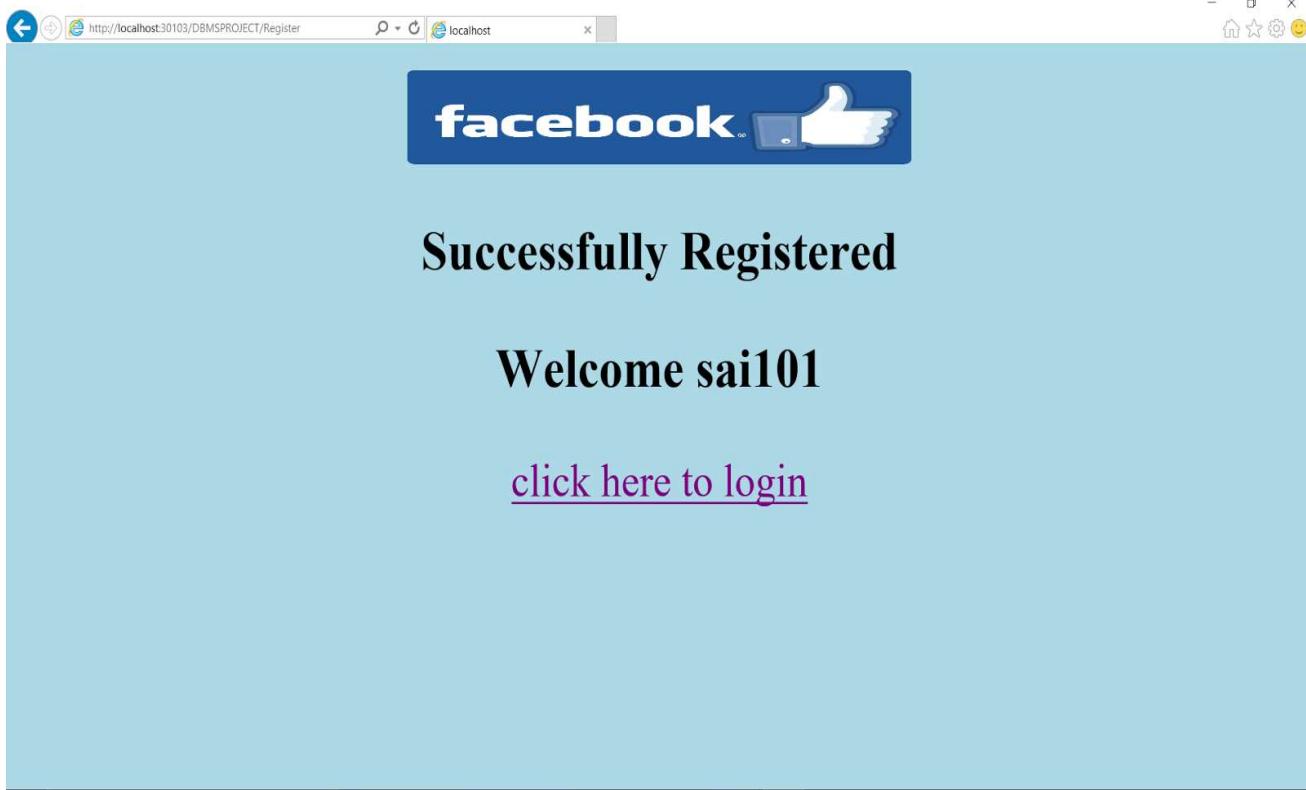


5.2: Registration Page

The screenshot shows a web browser window with the URL <http://localhost:30103/DBMSPROJECT/Register.html>. The page title is "Minifacebook Registration". The main content is a "Create a New Account" form. It features a large blue header bar with the word "facebook" and a thumbs-up icon. Below the header, the text "Create a New Account" and "Its free and always will be!" is displayed. The form includes four input fields: "Enter First name" (containing "sai"), "Enter last name" (containing "praveena"), "Enter username" (containing "saiT01"), and "Enter Password" (containing "*****"). There is also a "Date of Birth: MM/DD/YYYY" field containing "12/12/1990". Below these fields are two radio buttons for gender: "Male" (unchecked) and "Female" (checked). At the bottom right is a pink "Sign-Up" button.

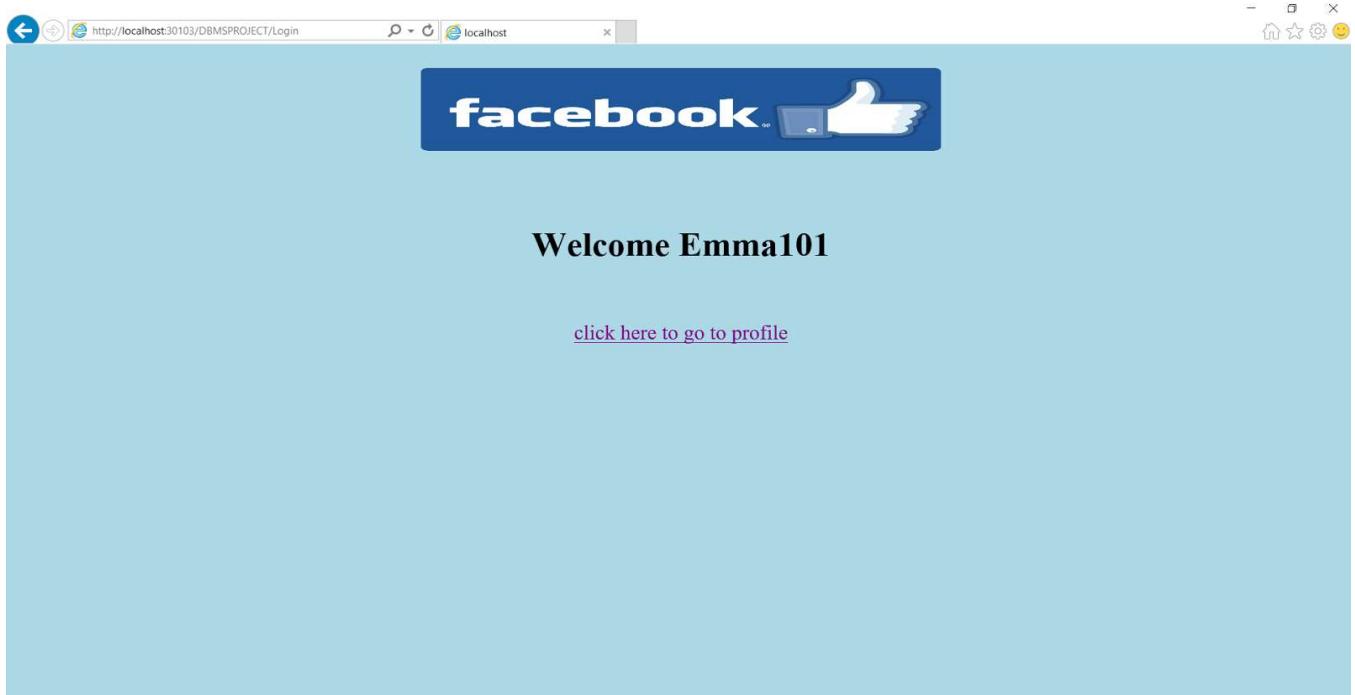
This screenshot shows the same registration page as the previous one, but with different data entered into the fields. The "First name" field now contains "sai", the "Last name" field contains "praveena", the "Username" field contains "saiT01", and the "Password" field contains "*****". The "Date of Birth" field still contains "12/12/1990". The gender selection remains the same, with "Female" checked. The "Sign-Up" button is visible at the bottom right.

5.3 Successfully Registered Page:

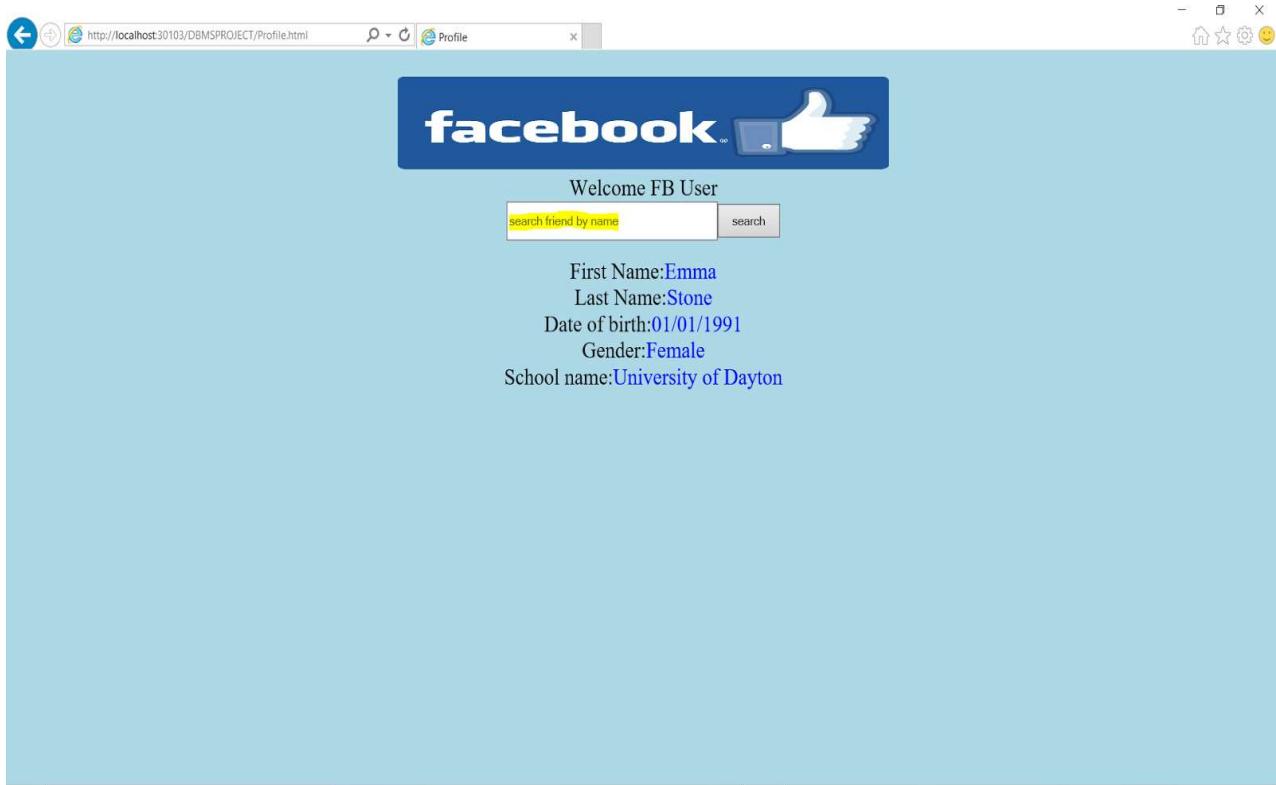


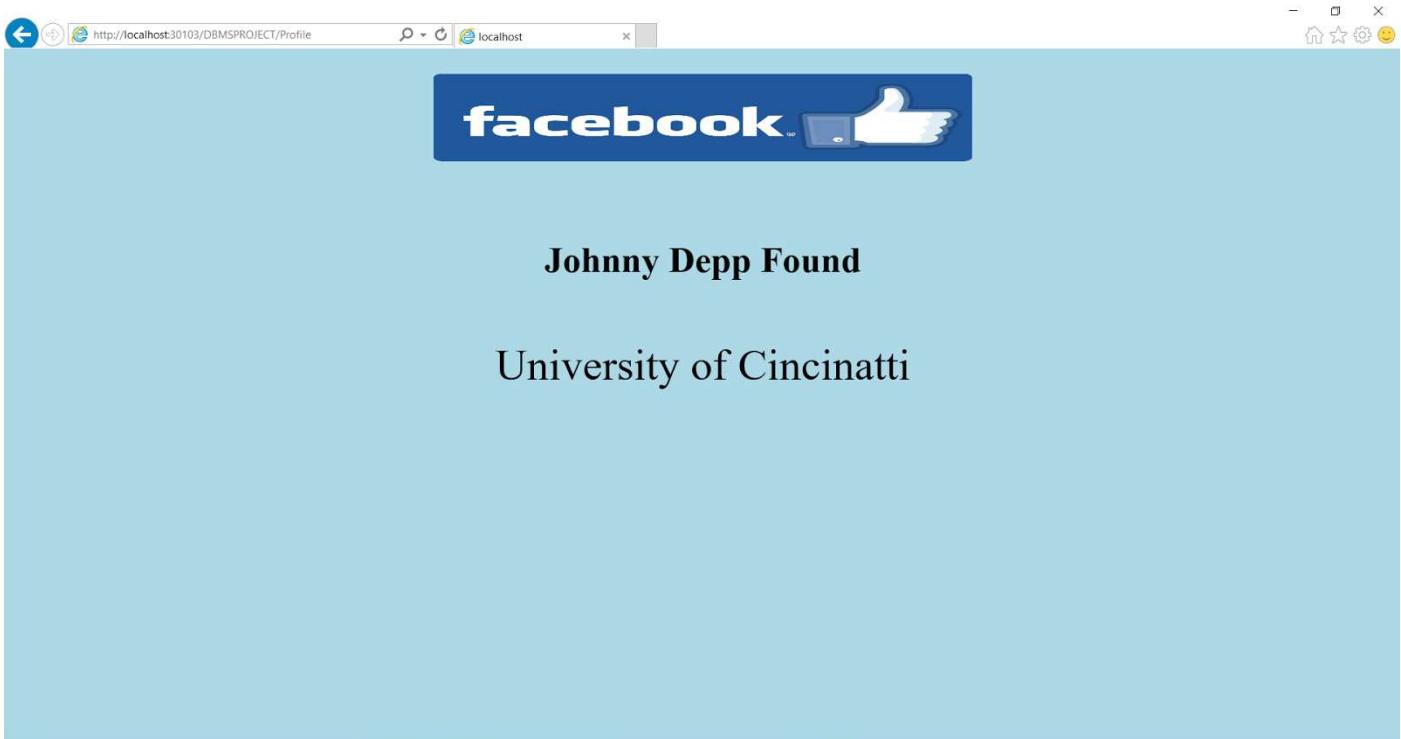
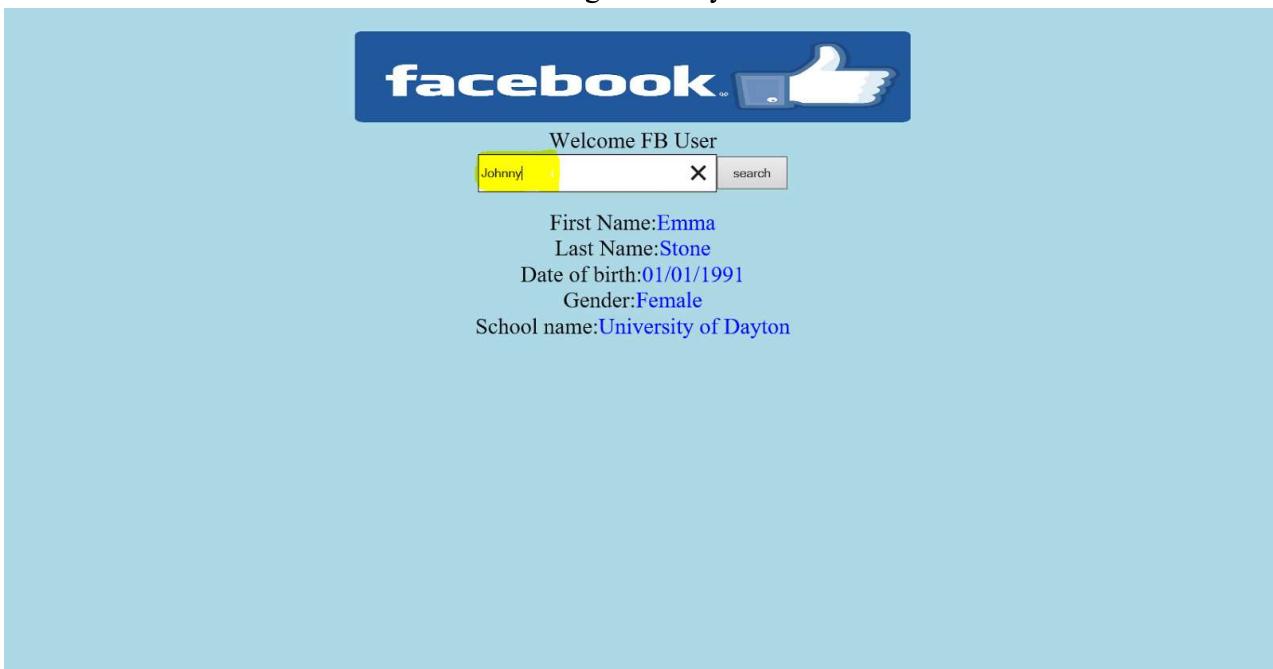
5.4 Welcome Page: After Entering the user Credentials





5.7: User Profile Page



5.5: Search Bar in Facebook: Searching friend by name

QUERY:

- Display the female users who are between the age group 18 and 25.**

```
SELECT user_id, fname||' '||Lname as Name, trunc((SYSDATE - BDATE)/365.25) as Age
from fb_user where trunc((SYSDATE - BDATE)/365.25) >= 18
and trunc((SYSDATE - BDATE)/365.25)<=25 and Gender='F' ;
```

USER_ID	NAME	AGE
106	Angelina Jolie	25
111	Kate Winslet	25
112	Scarlett Johansson	25

- Display the users who recently joined Facebook (after 2014)**

```
SELECT user_id, user_name
from fb_Account
where created_date>='01-Jan-2014';
```

USER_ID	USER_NAME
101	Emma101
102	Johnny102
105	Brda105
106	Angelina106
107	Leonardo107
109	Arnold109
111	Kate111
112	Scarlet112

- Display the users and count of their friend list from the Database**

```
SELECT f.user_id, u.fname||' '||u.Lname as Name,
Count(f.Friend_ID) as Num_of_Friends
FROM user_friend f,fb_user u
```

where u.user_id=f.user_id
 group by f.user_id,u.FNAME,u.lname;

USER_ID	NAME	NUM_OF_FRIENDS
101	Emma Stone	3
102	Johnny Depp	3
103	Al Pacino	3
104	Kevin Spacy	3
105	Brad Pitt	4

4. Display the different kinds of statuses updated by the Facebook users so far

SELECT unique(user_status) from user_status;

USER_STATUS
Feeling happy
Feeling annoying
Feeling motivated
Feeling loved
Feeling emotional
Feeling blessed
Feeling excited
Watching Transformers
Thinking about DMBS Project
Celebrating 25th Bday
Feeling sad
Feeling devotional

5. Create a new event for the user id 101

INSERT into Events Values (101,304,'Marathon','Be fit by participating in Marathon. Bring all your friends. We all will fun','30-Apr-2017');

```
1 row(s) inserted.
```

6. Display the events Emma Stone will be attending in the month of April 2017

```
SELECT event_Id, event_Name from Events where event_date>='01-Apr-2017' and event_date<='30-Apr-2017' and user_id=(select user_id from fb_user where fname='Emma' and lname='Stone');
```

EVENT_ID	EVEET_NAME
302	Indian Fest
303	Justin Bieber Concert
304	Marathon

7. Display all the Categories along with Category ID and Name which fall under Music category

```
SELECT Category_Id, Name from Category_Likes where Category_Type='Music';
```

CATEGORY_ID	NAME
401	Justin Bieber
402	Selena Gomez

8. Display the most visited place along with the city (more than 3 visits)

```
SELECT Location, city from Check_ins group by Location, city having Count(Location) >3;
```

LOCATION	CITY
Central Park	New York
Walt Disney World	Florida
Statue of Liberty	New York
Lincoln Memorial	Washington
Yellowstone National Park	Montana

9. Display the list of users who worked and currently working for Google

```
SELECT fname||' '||lname as Name from fb_user where user_id in (SELECT user_id from workplace where organization='Google');
```

LOCATION	CITY
Central Park	New York
Walt Disney World	Florida
Statue of Liberty	New York
Lincoln Memorial	Washington
Yellowstone National Park	Montana

10. Display the list of groups for which Johnny Depp is the Admin or owner of the group.

```
SELECT group_id,group_name from groups where owner_id in (select user_id from fb_user where Fname='Johnny' and Lname='Depp');
```

GROUP_ID	GROUP_NAME
602	Graveyord
606	Fantastic Family

11. Display the user id and username of the users who are not even owner of a single group

```
SELECT user_id,user_name from fb_user where user_id not in (select owner_id from groups);
```

USER_ID	USER_NAME
104	Kevin104
108	Tom108
109	Arnold109
110	Sylvester110
111	Kate111
112	Scarlet112

12. Display the usernames of users who are going to University of Dayton and who are in the groups 'University of Dayton' and 'Arts and Science';

```
SELECT user_name from fb_user where school_name like 'University of Dayton' and user_id in  
(select user_id from user_groups where group_name In ('University of Dayton','Arts and  
Science'));
```

USER_ID	USER_NAME
101	Emma101
104	Kevin104
107	Leonardo107
109	Arnold109

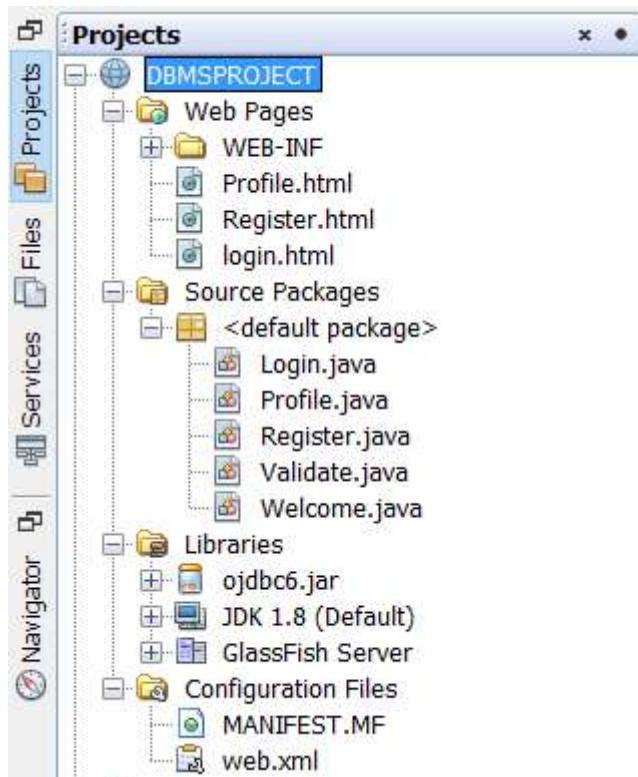
OBSERVATIONS:

The inspiration of our project is our Interest for Social Networking sites specially the Facebook.

Millions of people are connected using the facebook and it's fascinating to dream about having such a big database. Discussing with the group, we sat down together and we drew up a blueprint of what we thought our project needed and what information was critical. So came to a conclusion of having user, user friends, user likes, user albums, user status and user events which are very common these days. After we had this blue-print figured out we had a smooth transition in implementing our original mock up which included drawing the ER diagram, creating schema and populating the Database. We Hardly took a day to finish the Er-Diagram and the database schema. Glad that we took DBMS 1 class before registering for DBMS 2. Problem that we faced was at validating the pages. Which then we sorted by referring the previous java course that we have taken. All five of our team members were eager on implementing them and we were all trying to learn them as we went along with project. Overall our group learned a great deal from this project and it really helped our understanding of how much meticulous work and planning is needed just to setup a basic Database.

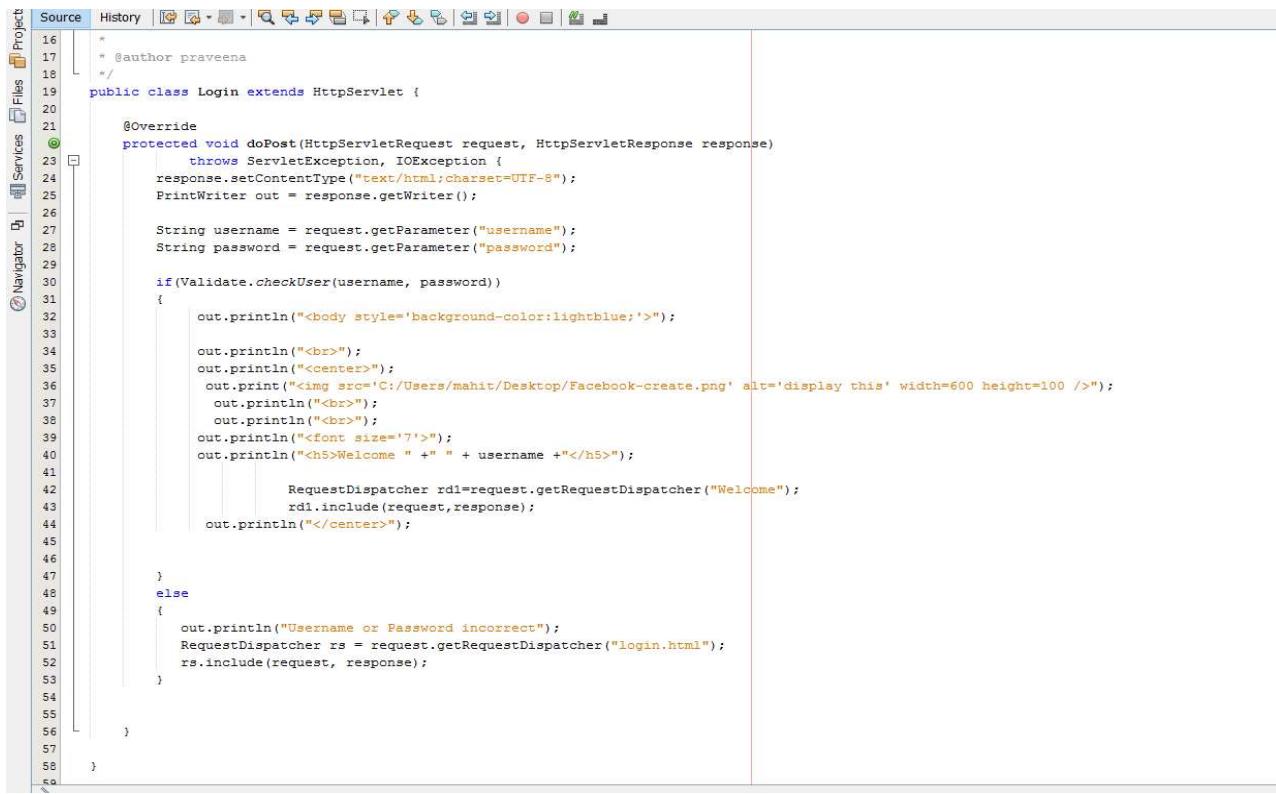
CODING

Project Explore:



LOGIN.PAGE: This file provides the main function for running the program. Username and the Password strings are been defined here. Validating the username and password. Using validate class which is been defined in validate.java class file. when you enter the username and password

in the login page it will display the login.html page if the credentials are valid or else it will show say that the username or password that you have entered is wrong.



The screenshot shows a Java IDE interface with the following details:

- Project:** A tree view showing a single project folder.
- Source:** The active tab, showing the code for a `Login` servlet.
- History:** A tab showing a history of changes.
- Files:** A tab showing a list of files.
- Services:** A tab showing a list of services.
- Navigator:** A tab showing a list of navigation items.
- Code Editor:** The main area containing the Java code for the `Login` servlet.

```
16  /*
17   * @author praveena
18   */
19  public class Login extends HttpServlet {
20
21      @Override
22      protected void doPost(HttpServletRequest request, HttpServletResponse response)
23          throws ServletException, IOException {
24          response.setContentType("text/html;charset=UTF-8");
25          PrintWriter out = response.getWriter();
26
27          String username = request.getParameter("username");
28          String password = request.getParameter("password");
29
30          if(Validate.checkUser(username, password))
31          {
32              out.println("<body style='background-color:lightblue;'>");
33
34              out.println("<br>");
35              out.println("<center>");
36              out.print("<img src='C:/Users/mahit/Desktop/Facebook-create.png' alt='display this' width=600 height=100 />");
37              out.println("<br>");
38              out.println("<br>");
39              out.println("<font size='7'>");
40              out.println("<h5>Welcome " + " " + username + "</h5>");
41
42              RequestDispatcher rd1=request.getRequestDispatcher("Welcome");
43              rd1.include(request,response);
44              out.println("</center>");
45
46          }
47      else
48      {
49          out.println("Username or Password incorrect");
50          RequestDispatcher rs = request.getRequestDispatcher("login.html");
51          rs.include(request, response);
52      }
53
54
55
56
57
58    }
59}
```

Profile.html

```

<!DOCTYPE html>
<html>
    <head>
        <title>Profile</title>
        <script language="JavaScript" type="text/JavaScript" ></script>
    <style>
        body
        {
            background-color: lightblue;
        }
    </style>
    </head>
    <body>
        <form method="post" action="Profile">
            <br>
            <center>
                
            <br>
            <font size="5">Welcome FB User</font>
            <br>
            <input type="text" style="height:35px; width:250px;" placeholder="search friend by name" name="fname" required/><input type="submit" style="height:35px; width:75px;" value="search" />
            <br>
            <br>
            <font size="5">First Name:</font><font size="5" color="Blue">Emma </font>
            <br>
            <font size="5">Last Name:</font><font size="5" color="Blue">Stone </font>
            <br>
            <font size="5">Date of birth:</font><font size="5" color="Blue">01/01/1991 </font>
            <br>
            <font size="5">Gender:</font><font size="5" color="Blue">Female </font>
            <br>
            <font size="5">School name:</font><font size="5" color="Blue">University of Dayton </font>
        </center>
    </body>
</html>

```

Profile.java This page is more about defining the user entities and extracting the data from the database using Prepared statements.

```


* @author praveena
*/
public class Profile extends HttpServlet {
protected void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
    response.setContentType("text/html;charset=UTF-8");
    PrintWriter out = response.getWriter();
    boolean st =false;
    try{
        Class.forName("oracle.jdbc.OracleDriver");
        Connection con=DriverManager.getConnection
        ("jdbc:oracle:thin:@localhost:1521:XE","SYSTEM","admin");
        String fname = request.getParameter("fname");

        PreparedStatement ps=con.prepareStatement("select * from PRAVEENA.fb_user where fname=?");
        ps.setString(1, fname);

        ResultSet rs=ps.executeQuery();

        while(rs.next()){
            String firstname=rs.getString("fname");
            String lastname =rs.getString("lname");
            //out.println("hiiiiii");

            String school=rs.getString("school_name");
            String sex=rs.getString("gender");
            out.println("<body style='background-color:lightblue;'>");

            out.println("<br>");
            out.println("<center>");
            out.print("<img src='C:/Users/mahit/Desktop/Facebook-create.png' alt='display this' width=600 height=100 />");
            out.println("<br>");
            out.println("<br>");
            out.println("<font size='7'>");
            out.println("<h5> " +firstname +" "+ lastname + " " +"Found</h5>");
            out.println("<h5>School: " + " " + school + "</h5>");

            out.println("<br>");

```

Register.java : Fetching the User Information from database and validating the Registration user details. The class contains parameterized names for the tables you need to use in the queries, constructed in the class structure. This class implement the query functions.

Source History

```
public class Register extends HttpServlet {
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();
        int user_id=113;

        String fname = request.getParameter("fname");

        String Lname=request.getParameter("lname");
        String username = request.getParameter("username");
        String password = request.getParameter("password");

        String birthday = request.getParameter("dob");
        String gender=request.getParameter("sex");

        try{
            Class.forName("oracle.jdbc.OracleDriver");
            Connection con=DriverManager.getConnection
                ("jdbc:oracle:thin:@localhost:1521:XE","SYSTEM","admin");
            PreparedStatement ps=con.prepareStatement
                ("insert into PRAVEENA.fb_user values(?,?,?,?,?,?)");
            ps.setInt(1,user_id);
            ps.setString(2, fname);
            ps.setString(3, Lname);
            ps.setString(4, username);
            ps.setString(5, password);
            ps.setString(6, birthday);
            ps.setString(7,gender);
            ps.setString(8, null);
            int i=ps.executeUpdate();

            if(i>0)
            {
                String h3;
                out.println("<body style='background-color:lightblue;'>");
                out.println("<br>");
                out.println("<center>");
                out.print("<img src='C:/Users/mahit/Desktop/Facebook-create.png' alt='display this' width=600 height=100 />");
                out.println("</center>");
            }
        }catch(Exception e){e.printStackTrace();}
    }
}
```

Register.html

```
<!DOCTYPE html>
<html>
<head>
<title>MiniFacebook Registration</title>
<script language="JavaScript" type="text/JavaScript" ></script>
<style>
body
{
background-color: lightblue;
}
</style>
</head>
<body>
<form method="post" action="Register">
<br>
<center>

<br>
<br>
<center>
<font size="12">Create a New Account</font>
<br><br>
<font size="6">Its free and always will be!</font>
<br><br>
</center>
<input type="text" style="height:35px; width:150px;" placeholder="Enter First name" name="fname" required/>
&ampnbsp&ampnbsp
<input type="text" style="height:35px; width:150px;" placeholder="Enter last name" name="lname" required/>
<br><br>
<input type="text" style="height:35px; width:150px;" placeholder="Enter username" name="username" required/>
<br><br>
<input type="password" style="height:35px; width:150px;" placeholder="Enter Password" name="password" required/>
<br><br>
Date of Birth: <input type="date" value="" name="dob" required/><br><br>
<br>
<input type="radio" name="sex" value="M">Male &ampnbsp&ampnbsp
<input type="radio" name="sex" value="F">Female<br>
<br><br>
<input type="submit" style="height:35px; width:150px;" value="Sign-Up"/>
</form>
</body>
</html>
```

Login.java

```
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
public class Login extends HttpServlet {
    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();
        String username = request.getParameter("username");
        String password = request.getParameter("password");
        if(Validate.checkUser(username, password))
        {
            out.println("<body style='background-color:lightblue;'>");
            out.println("<br>");
            out.println("<center>");
            out.print("<img src='C:/Users/mahit/Desktop/Facebook-create.png' alt='display this' width=600 height=100 />");
            out.println("<br>");
            out.println("<br>");
            out.println("<font size='7'>");
            out.println("<h5>Welcome " +" " + username +"</h5>");
            RequestDispatcher rd1=request.getRequestDispatcher("Welcome");
            rd1.include(request,response);
            out.println("</center>")
        }
        else
        {
            out.println("Username or Password incorrect");
            RequestDispatcher rs = request.getRequestDispatcher("login.html");
            rs.include(request, response);
        }}}
```

Validate.java

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.Properties;
import java.util.*;
import javax.servlet.*;
public class Validate {
    public static boolean checkUser(String username, String password)
    {
        boolean st = false;
        try{
            //loading drivers for mysql
            Class.forName("oracle.jdbc.OracleDriver");
            //creating connection with the database
            Connection con=DriverManager.getConnection
                ("jdbc:oracle:thin:@localhost:1521:XE","SYSTEM","admin");
            PreparedStatement ps=con.prepareStatement("select * from TASNEEM.fb_user where
            user_name=? and password=?");
            ps.setString(1, username);
            ps.setString(2, password);
            ResultSet rs=ps.executeQuery();
            if(rs.next())
            {
                st = true;
            }
        return st;
    } catch (Exception e)
    {
        e.printStackTrace();
    }
    return false;
} }
```

Welcome.java

```
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
public class Welcome extends HttpServlet {
    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();
        out.println("<font size='5'>");
        out.print("<a href=\"" + getServletContext().getContextPath() + "/Profile.html">click here to go
to profile</a>");
    }
}
```

Register.java

```
import static com.sun.org.apache.xalan.internal.lib.ExsltDatetime.date;
import java.io.IOException;
import java.io.PrintWriter;
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.sql.*;
import java.text.DateFormat;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.util.Date;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.servlet.GenericServlet;
```

```
public class Register extends HttpServlet {  
    protected void doPost(HttpServletRequest request, HttpServletResponse response)  
        throws ServletException, IOException {  
        response.setContentType("text/html;charset=UTF-8");  
        PrintWriter out = response.getWriter();  
        int user_id=113;  
        String fname = request.getParameter("fname");  
        String Lname=request.getParameter("lname");  
        String username = request.getParameter("username");  
        String password = request.getParameter("password");  
        String birthday = request.getParameter("dob");  
        String gender=request.getParameter("sex");  
        try{  
            Class.forName("oracle.jdbc.OracleDriver");  
            Connection con=DriverManager.getConnection  
                ("jdbc:oracle:thin:@localhost:1521:XE","SYSTEM","admin");  
            PreparedStatement ps=con.prepareStatement  
                ("insert into TASNEEM.fb_user values(?,?,?,?,?,?)");  
            ps.setInt(1,user_id);  
            ps.setString(2, fname);  
            ps.setString(3, Lname);  
            ps.setString(4, username);  
            ps.setString(5, password);  
            ps.setString(6, birthday);  
            ps.setString(7,gender);  
            ps.setString(8, null);  
            int i=ps.executeUpdate();  
            if(i>0)  
            {  
                String h3;  
                out.println("<body style='background-color:lightblue;'>");  
                out.println("<br>");  
                out.println("<center>");  
                out.print("<img src='C:/Users/mahit/Desktop/Facebook-create.png' alt='display this'  
width=600 height=100 />");  
                out.println("<br>");  
                out.println("<font size='7'>");  
                out.println("<h3>Successfully Registered</h3>");  
                out.println("<h3>Welcome " +" "+ username +"</h3>");  
            }  
        }  
    }  
}
```

```
        out.print("<a href=\"" + getServletContext().getContextPath() + "/login.html">click here to  
login</a>");  
        out.println("</center>");  
    }  
    else  
    {  
        out.println("hello");  
    }  
    catch(Exception se)  
{  
    se.printStackTrace();  
}}
```

Profile.java

```
import java.io.IOException;  
import java.io.PrintWriter;  
import javax.servlet.ServletException;  
import javax.servlet.http.HttpServlet;  
import javax.servlet.http.HttpServletRequest;  
import javax.servlet.http.HttpServletResponse;  
import java.sql.Connection;  
import java.sql.DriverManager;  
import java.sql.PreparedStatement;  
import java.sql.ResultSet;  
import java.sql.SQLException;  
import java.sql.Statement;  
import java.util.Properties;  
import java.util.*;  
import javax.servlet.RequestDispatcher;  
  
public class Profile extends HttpServlet {  
    protected void doPost(HttpServletRequest request, HttpServletResponse response)  
        throws ServletException, IOException {  
        response.setContentType("text/html;charset=UTF-8");  
        PrintWriter out = response.getWriter();  
        boolean st = false;  
        try {  
            Class.forName("oracle.jdbc.OracleDriver");  
            Connection con = DriverManager.getConnection
```

```
("jdbc:oracle:thin:@localhost:1521:XE","SYSTEM","admin");
String fname = request.getParameter("fname");

PreparedStatement ps=con.prepareStatement("select * from TASNEEM.fb_user where
fname=?");
ps.setString(1, fname);
ResultSet rs=ps.executeQuery();
while(rs.next()){
String firstname=rs.getString("fname");
String lastname =rs.getString("lname");
//out.println("hiiii");
String school=rs.getString("school_name");
String sex=rs.getString("gender");
out.println("<body style='background-color:lightblue;'>");
out.println("<br>");
out.println("<center>");
out.print("<img src='C:/Users/mahit/Desktop/Facebook-create.png' alt='display this' width=600
height=100 />");
out.println("<br>");
out.println("<br>");
out.println("<font size='7'>");
out.println("<h5> " +" "+firstname +" "+ lastname + " " +"Found</h5>");
out.println("<h5School:> " +" "+ school +"</h5>");
if(firstname.isEmpty()||lastname.isEmpty()||school.isEmpty()||sex.isEmpty())
{
    RequestDispatcher rd = request.getRequestDispatcher("Profile.html");

        rd.include(request, response);
    }}})
catch(Exception se)
{
    se.printStackTrace();
}}}
```

OBSERVATIONS:

The inspiration of our project is our Interest for Social Networking sites specially the Facebook.

Millions of people are connected using the facebook and it's fascinating to dream about having such a big database. Discussing with the group, we sat down together and we drew up a blueprint of what we thought our project needed and what information was critical. So came to a conclusion of having user, user friends, user likes, user albums, user status and user events which are very common these days. After we had this blue-print figured out we had a smooth transition in implementing our original mock up which included drawing the ER diagram, creating schema and populating the Database. We Hardly took a day to finish the Er-Diagram and the database schema. Glad that we took DBMS 1 class before registering for DBMS 2. Problem that we faced was at validating the pages. Which then we sorted by referring the previous java course that we have taken. All five of our team members were eager on implementing them and we were all trying to learn them as we went along with project. Overall our group learned a great deal from this project and it really helped our understanding of how much meticulous work and planning is needed just to setup a basic Database.