**ITCS 6160-091/8160-091**

**Database Systems**

**Final Report**

**Team Members**

1. **Naga Bijesh Roy Raya - #800846698**
2. **Sabari Trinadh Reddy Tippareddy - #800856078**
3. **RohanBendre - #800857096**
4. **KashyapBadani - #800846299**
5. **Tejavardhana chary Viswakarma - #800884268**
6. **PraharashaKorupolu - #800858193**
7. **PraneethMadhav Chanda - #800866867**

**Submitted to:**

**Dr. Pamela Thompson**

**plthomps@uncc.edu**

**KDD Lab Woodward Hall**

**Department of Computer Science**

**University of North Carolina at Charlotte**

**Charlotte**

**Team Structure**

Naga Bijesh Roy Team Leader 513-641-7605 [nagabijeshroy@gmail.com](mailto:nagabijeshroy@gmail.com) Business Logic

RohanBendre Team 407-923-3570 [rohanbendre@gmail.com](mailto:rohanbendre@gmail.com) Secretary/Scribe/DBDesigner

Trinadh Reddy Team Member 937-813-9229 [trinadh.tstr@gmail.com](mailto:trinadh.tstr@gmail.com) Business Logic

KashyapBadani Team Memeber 980-267-0004 [badani.kashyap7@gmail.com](mailto:badani.kashyap7@gmail.com) UML Design / Business Logic

PraneethMadhav Team Memeber 980-636-7311 [praneethmadhav@gmail.com](mailto:praneethmadhav@gmail.com) GUI Designer

PraharshaKorupolu Team Member 352-238-4832 [kpraharsha@gmail.com](mailto:kpraharsha@gmail.com) GUI Designer

Tejavardhana Chary Team Memeber 980-833-3990 [tejavardhanchary@gmail.com](mailto:tejavardhanchary@gmail.com) DB Designer

**Meeting Schedule**

Monday 9:15-10 am

Wednesday 9:15-10 am

Friday 10-1 pm

**Modes of Communication:**

Google Drive-https://drive.google.com/#folders/0B7LlZvnonYHcV3FwbGhvY1lyYVk

Facebook - Team Connect

Whatsapp - Team Connect

**Project Plan with Tentative Schedule for completion of the Project:**

User Requirements - September 8th, 2014

Design of UML Diagrams - September18th, 2014

Database Design - October 1st, 2014

GUI Design- Common Template Design - October 8th, 2014

Module Development using PhP as a Scripting - Nov 2nd, 2014

Language

Integrating all the Modules - Nov 14th, 2014

Testing - November 24th, 2014

Project Delivery/Project Demo Presentation - November 29th, 2014

**Project Description:**

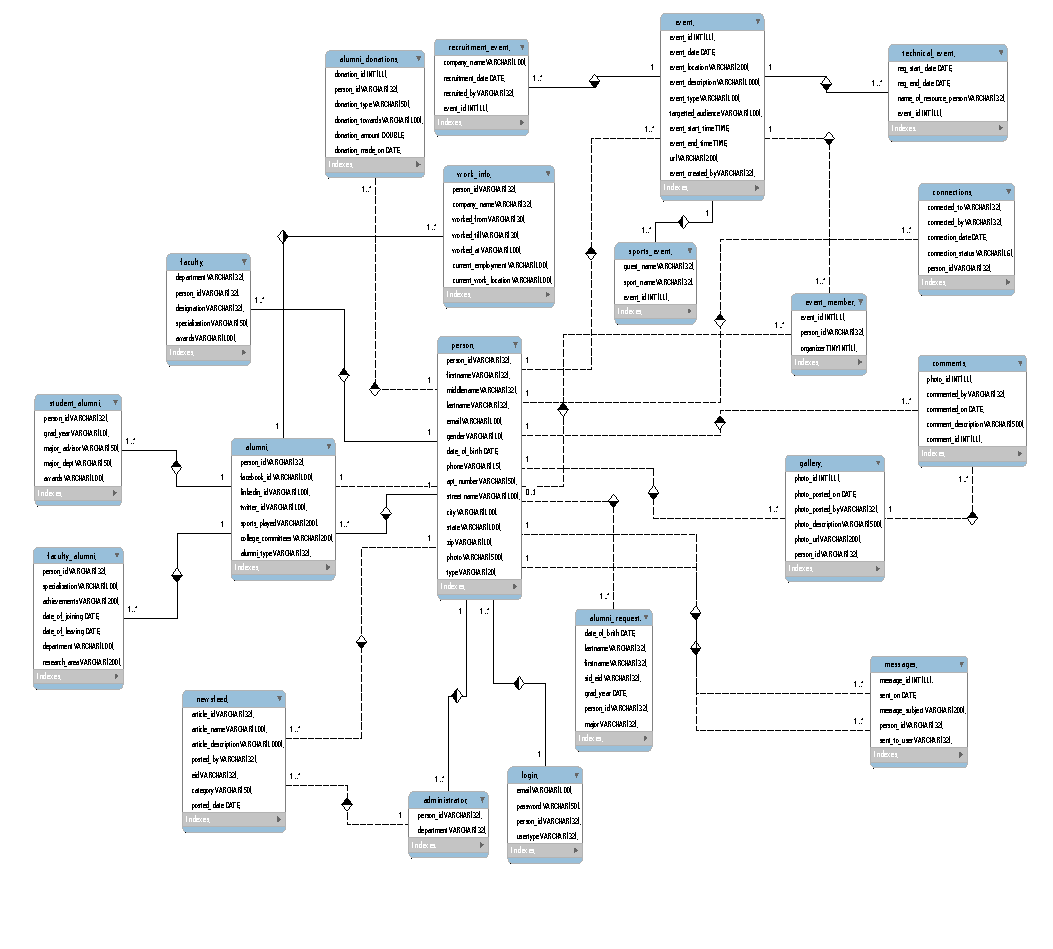
It is of utmost importance for a university to keep track of the information of their alumni. An alumni association of any university forms an immense network of professionals. It is possible for the current students to get a foot in the door for next job if they make efficient use of the alumni contacts. Also, for the students who seek better career opportunities, using the alumni directory to connect with people who could be great resources for informational interviews or introductions is of use. University can utilize their alumni association to raise funds for the development of its infrastructure as well as providing funds for the meritorious students.

At the same time, it is essential to maintain database of the alumni through a web portal in order to ease the task of connecting all the alumni of the university together. This project will develop a new commercial alumni connection system that would keep track of the information of entire alumni along with several business related functionalities to be incorporated.

These functionalities include offering donations to the university, creating events among the alumni through the portal and recruitment for the current students of the university. In addition to that, this web portal will give alumni a method to establish connections among themselves by sending invites, which let them, maintain contacts through a messaging service which is a key feature of the proposed system. Also the web portal that we propose to develop gives faculty the ability to search alumni based on their geography as well as the industry they work at. Professors of the university can connect with the alumni based on the alumni major and their graduation years.

The development of web portal for University alumni system will definitely solve the problem faced by most of the universities by enabling efficient alumnus management.

**EER Diagram in MySQL Work bench:**



**Narrative:**

**Entities:**

1. **Person(PK person\_id)** – This is the superclass and has subclasses as faculty, alumni, admin.
2. **Alumni (PK person\_id)** – This is the subclass of person entity and has student alumni and faculty alumni as subclasses.
3. **Alumni Donations** (PK DonationID) – The alumni can make donations to university.
4. **Gallery** (PK PhotoID) – The alumni can add photos to the library.
5. **Comments** (PK CommentID) – Alumni can comment on the photos that have been added.
6. **Faculty** (PK EID) – Faculty can request information about alumni and events.
7. **Connections** (PK ConnectionID{ConnectedTo,ConnectedBy}) – Alumni will have connections with different alumni.
8. **Login** (PK UserID) – Every alumni will have to login to use the application.
9. **Messages** (PK Message ID) – Alumni can send messages to each other.
10. **Student** (PK StudentID) – Student can view alumni
11. **Event** (PK EventID) – Event is a superclass and has subclasses as SportsEvent, TechEvent, RecruitmentEvent.
12. **Event Member –** Used to keep details of people participating in events and details of event organizer.
13. **Alumni Request** (PK RequestID{firstname,LastName,DateOfBirth}) – Alumni makes requests.
14. **Administrator** (PK EID) – Administrator approves requests made by alumni.
15. **News Feed** (PK ArticleID) – The news about different events will be posted on news feed.

**Relationships:**

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Entity Name** | **Relationship** | **Entity Name** |
| 1 | Alumni | Gives | Alumni Donations |
| 2 | Alumni | Attends | Event |
| 3 | Alumni | Has | Login |
| 4 | Alumni | Has | Connections |
| 5 | Alumni | Sends | Messages |
| 6 | Alumni | Has | Gallery |
| 7 | Alumni | Gives | Comments |
| 8 | Gallery | Has | Comments |
| 9 | Faculty | Views | Alumni |
| 10 | Faculty | Has | Login |
| 11 | Administrator | Post News | News Feed |
| 12 | Alumni | Request Invite | Administrator |
| 13 | Alumni Request | Request | Administrator |
| 14 | Student | Views | Alumni |
| 15 | Student | Views | News Feed |
| 16 | Administrator | Approves | Event |
| 17 | Faculty | Requests Event | Administrator |
| 18 | Person | Goes to | Event Member |

**EER Description:**

The EER diagram depicts the database design for our University Alumni Connection System. The main functionalities of this application include creation of an alumni account, modify profile, donating funds to college, connecting with other alumni, creating an event, and updating news feed.

The proposed system starts with the registration of alumni (Student/Faculty) after he/she completes his/her graduation from the university. Alumni registers to this application by providing his basic information such as Name, Date of Birth, Student id/ employee id, Username, Year of Graduation and Major. If alumni forgets their University Id, which is mandatory for creating an account they need to send a request to the administrator of this system.

**Person Entity** is further specialized into Faculty, Alumni and Admin. Alumni entity is specialized into student alumni and faculty alumni.

Faculty Alumni has the specialization, achievements, date of joining, date of leaving, department, and research area as the attributes. Whereas Student alumni have graduation year, major advisor, major department, awards as the attributes along with the basic details.

Alumni Entity has College clubs, Sports played and College Committees as multi valued attributes, which are separated by commas.

After creation of Account, alumni can update the profile mentioning their address, work information (current and previous employer), contact information (email and phone number), social account details (Facebook, LinkedIn, and Twitter), profile photo, nationality, short introduction, occupation type, and extracurricular activities. Faculty alumni also need to update specialization, achievements, last working date, department, date of joining and research area. Student alumni should update year of graduation, awards and recognitions, major department and advisor name.

Alumni will be able to request for an event invite providing event description, event start and end date/time, location and targeted audience. Events can be categorized as sports event (Sport name & guest name), tech event (Registration deadlines, name of resource person) and recruitment event (organization name, recruitment date, recruited by). Once the request is submitted, Administrator has the privilege to approve or deny the event invitation. After the approval, the event details are updated in news feed. Every post is considered an article in news feed which can be uniquely identified by article id.

Alumni will also be able to make donation. Each donation will have details like, amount donated, donation date, donation towards and type of donation.

Connection information between alumni is identified based on connection status, connection to, connected by and connection date. Alumni will be able to communicate through messages. Each message will have message Id, timestamp, message, sender and receiver.

Alumni can create a gallery and upload photos, which has attributes namely photo description, photo id, posted date, and username. Also alumni can post comments on the posted photos, which have attributes as photo id, comment description, comment id, commented by and commented on date.

**Queries with description to access the tables, insert and update the records:**

**Search Alumni:**

All the alumni details will be stored in alumni table. One can search the details of alumni based on different fields like firstname, lastname, username, email, phone\_number, sports\_played and college\_committees. Few of the queries that are used to view the alumni table is shown below:

The query used when user searches based on firstname or lastname or DOB

**Select \* from alumni a inner join person p on a.person\_id=p.person\_id**

**wherep.firstname = 'Teja' or p.lastname = 'Chary' or p.date\_of\_birth = '1991-07-03';**

The query used when user searches based on firstname and lastname or DOB

**Select \* from alumni a inner join person p on a.person\_id=p.person\_id**

**wherep.firstname = 'Teja' and p.lastname = 'Chary' or p.date\_of\_birth ='1991-07-03';**

The query used when user searches based on firstname or lastname and DOB

**Select \* from alumni a inner join person p on a.person\_id=p.person\_id**

**wherep.firstname = 'Teja' or p.lastname = 'Chary' and p.date\_of\_birth ='1991-07-03';**

The query used when user searches based on firstname and lastname and DOB

**Select \* from alumni a inner join person p on a.person\_id=p.person\_id**

**wherep.firstname = 'Teja' and p.lastname = 'Chary' and p.date\_of\_birth ='1991-07-03';**

Query used when user searches based on person\_id

**Select \* from alumni a inner join person p on a.person\_id=p.person\_id**

**Where a.person\_id = 1234;**

Query used when user searches based on firstname or lastname or city or state

**Select \* from alumni a inner join person p on a.person\_id=p.person\_id**

**wherep.firstname = 'Teja' or p.lastname = 'Chary' or p.city= 'Charlotte' or p.state = 'North Carolina';**

Query used when user searches based on firstname and lastname or city or state

**Select \* from alumni a inner join person p on a.person\_id=p.person\_id**

**wherep.firstname = 'Teja' and p.lastname = 'Chary' or p.city= 'Charlotte' or p.state = 'North Carolina';**

Query used when user searches based on firstname or lastname and city or state

**Select \* from alumni a inner join person p on a.person\_id=p.person\_id**

**wherep.firstname = 'Teja' or p.lastname = 'Chary' and p.city= 'Charlotte' or p.state = 'North Carolina';**

Query used when user searches based on firstname or lastname or city and state

**Select \* from alumni a inner join person p on a.person\_id=p.person\_id**

**wherep.firstname = 'Teja' or p.lastname = 'Chary' or p.city= 'Charlotte' and p.state = 'North Carolina';**

Query used when user searches based on firstname and lastname and city or state

**Select \* from alumni a inner join person p on a.person\_id=p.person\_id**

**wherep.firstname = 'Teja' and p.lastname = 'Chary' and p.city= 'Charlotte' or p.state = 'North Carolina';**

Query used when user searches based on firstname or lastname or city and state

**Select \* from alumni a inner join person p on a.person\_id=p.person\_id**

**wherep.firstname = 'Teja' or p.lastname = 'Chary' or p.city= 'Charlotte' and p.state = 'North Carolina';**

Query used when user searches based on firstname and lastname and city and state

**Select \* from alumni a inner join person p on a.person\_id=p.person\_id**

**wherep.firstname = 'Teja' and p.lastname = 'Chary' and p.city= 'Charlotte' and p.state = 'North Carolina';**

**To view the profile:**

Query used to view stuent\_alumni profile based on alumni search

**Select \* from alumni as a, student\_alumni as sa join a.username=sa.username and a.username=’teja’;**

Query used to view faculty\_alumni profile based on alumni search

**Select \* from alumni as a, faculty\_alumni as fa join a.username=fa.username and a.username=’kashyap’;**

**Connection Invitations**

After searching the alumni, to connect to that alumni connection invite is send. When the connection invites is send data is inserted into connections table.

**INSERT INTO university\_alumni.connections (connected\_to, connected\_by, connection\_date, connection\_status, person\_id) VALUES ('bijesh', 'kashyap', '2013-01-20', 'inactive', ‘800846299');**

Where ‘connected\_to’ field specifies the alumni to whom request is send and ‘connected\_by’ is the one who sends the request. When the request is sent the ‘connection\_status’ is inactive.

The alumni can view his connection invites and can accept or reject the invites

**select connected\_by as 'Connection Invites', connection\_date as 'Requested On' from connections where connected\_to='trinadh';**

After the request is accepted by other alumni connection status is changed to active.

**update university\_alumni.connections set connection\_status='active' where connected\_to='rohan' and connected\_by='kashyap';**

**Gallery & Comments**

All the alumni members and staff will have a personalisedgallery.New images can be inserted ,existing images can be deleted in the gallery.Each image in the gallery will have image description and posting date.All the connections of alumni can comment on the images uploaded and at the same time previously posted comments can be edited/deleted.

**Queries:**

**Insert image to gallery**

**INSERT INTO ` university\_alumni`.`gallery` (`photo\_id`, `photo\_posted\_on`, `photo\_posted\_by`, `photo\_description`, `photo\_url`,`person\_id` ) VALUES ('345', '2010-12-12', 'Trinadh Reddy', 'India Trip','C:/Alumni/trinadh/345', '800856078');**

**Delete image from gallery (Image and comments assosciated with the image are deleted)**

**DELETE FROM ` university\_alumni`.`gallery`,` university\_alumni`.`comments`**

**WHERE photo\_id="345";**

**Insert Comments to image**

**INSERT INTO `university\_alumni`.`comments` (`photo\_id`, `commented\_by`, `commented\_on`, `comment\_description`, `comment\_id`) VALUES ('345', 'Raju', '2010-06-27', 'Scenery Comment', '111');**

**Edit comments for an image**

**UPDATE `university\_alumni`.`comments`**

**SET comment\_description="Good Image!! ", commented\_on=CURRENT\_TIMESTAMP**

**WHERE comment\_id= 111 ;**

**Delete comments for an image**

**DELETE FROM `university\_alumni`.`comments`**

**WHERE comment\_id= 111;**

**Donation by alumni:**

The alumni will search for all the events that will be taking place for a specified date range. He can choose to donate towards whichever event he wants to. Index is created on (event\_date, event\_description,event\_type) for quick and efficient retreival of events. The query for creating index is :

**create unique index retrieve\_event on event(event\_date,event\_description,event\_type);**

The query for searching events is :

**selectevent\_date,event\_description,event\_type,targetted\_audience from event where event\_date between ‘yy-mm-dd’ and ‘yy-mm-dd’;**

When alumni makes a donation, entry will be added to database. The query for inserting into database is :

**INSERT INTO `university\_alumni`.`alumni\_donations` (`donation\_id`, `person\_id`, `donation\_type`, `donation\_towards`, `donation\_amount`, `donation\_made\_on`) VALUES ('1235', ‘800857096’, 'cash', 'internships', '3000', '2014-10-08');**

**Profile creation by Administrator:**

Administrator creates an account for each alumni up on their request. Each account consists of these details :

         (person\_id,firstname,lastname,email,gender,date\_of\_birth and type) and these could not be modified.

**INSERT INTO `university\_alumni`.`person` (`person\_id`, `firstname`, `middlename`, `lastname`, `email`, `gender`, `date\_of\_birth`, `type`) VALUES ('800866867', 'Praneeth Madhav', NULL, 'Chanda', 'pchanda1@uncc.edu', 'Male', '1989-12-13', NULL, 'student');**

**INSERT INTO `university\_alumni`.`person` (`person\_id`, `firstname`, `middlename`, `lastname`, `email`, `gender`, `date\_of\_birth`, `type`) VALUES ('800812912', 'Bijesh', NULL, 'Roy', 'bijesh@uncc.edu', 'Male', '1991-04-12', NULL, 'faculty');**

**INSERT INTO `university\_alumni`.`person` (`person\_id`, `firstname`, `middlename`, `lastname`, `email`, `gender`, `date\_of\_birth`, `type`) VALUES ('800856856', 'Trinadh', NULL, 'Reddy', 'Trinadh@uncc.edu', 'Male', '1992-05-12', NULL, 'adminstrator');**

**Profile update by Alumni:**

Alumni can update his profile :

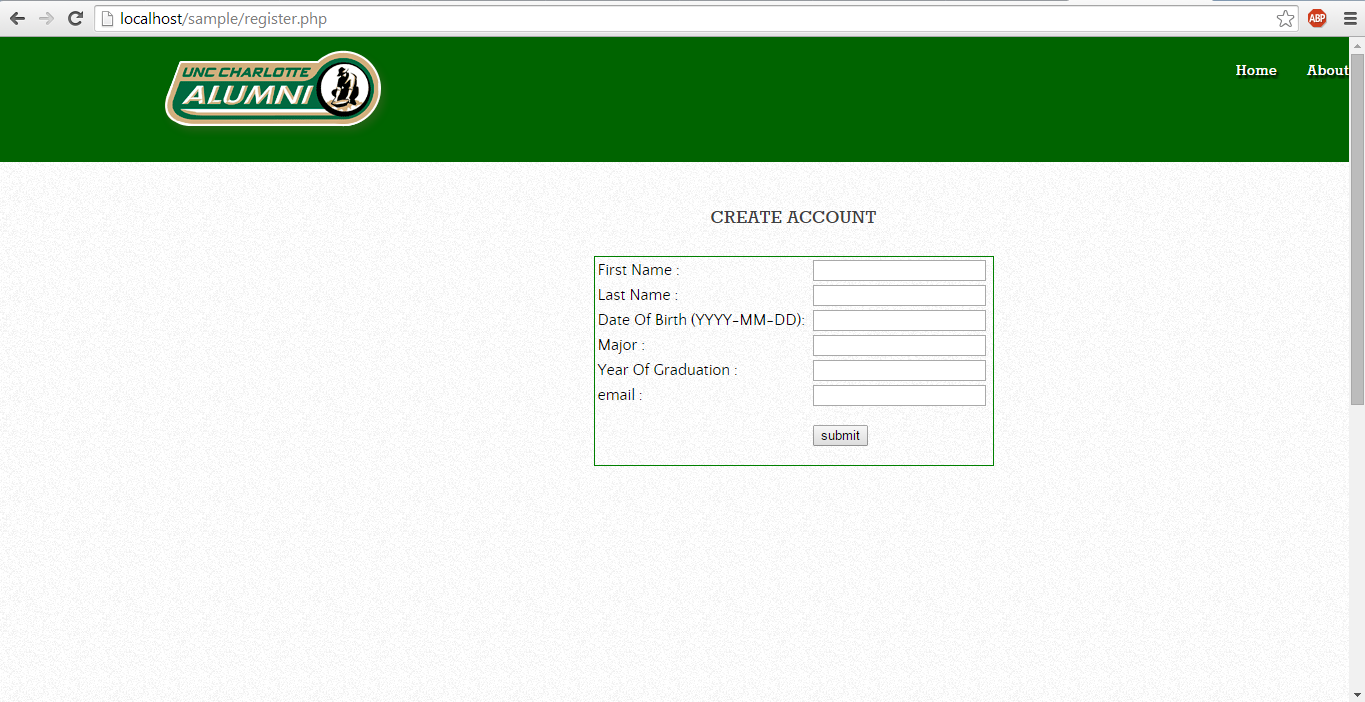
**UPDATE `university\_alumni`.`person` SET `phone` = '9806367311', `apt\_number` = '9551 university Terrace', `city` = 'Charlotte', `state` = 'North Carolina', `zip` = '28262' WHERE `person`.`person\_id` = '800866867';**

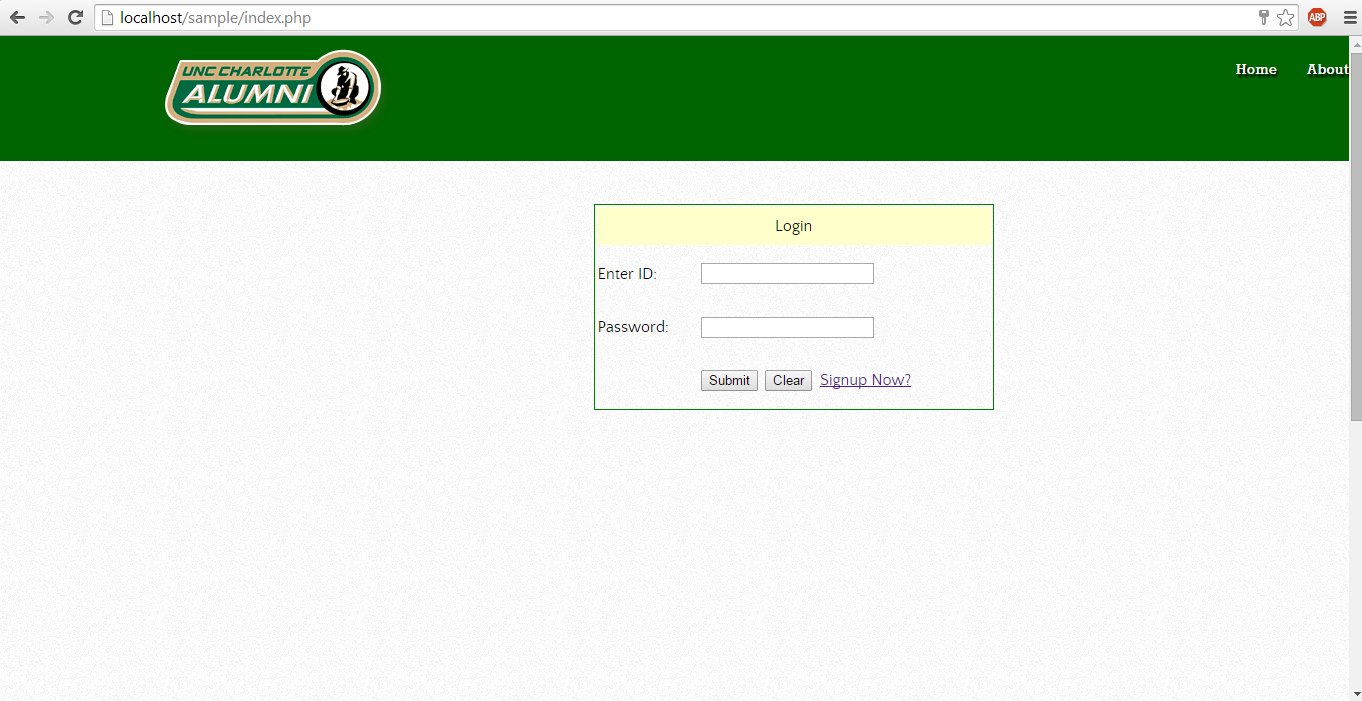
**INSERT INTO `university\_alumni`.`alumni` (`person\_id`, `facebook\_id`, `linkedin\_id`, `twitter\_id`, `sports\_played`, `college\_committees`, `alumni\_type`) VALUES ('800866867', 'praneeth@fb.com', 'praneeth@linkedin.com', 'praneeth@twitter.com', 'Football', '49th security divison', 'student');**

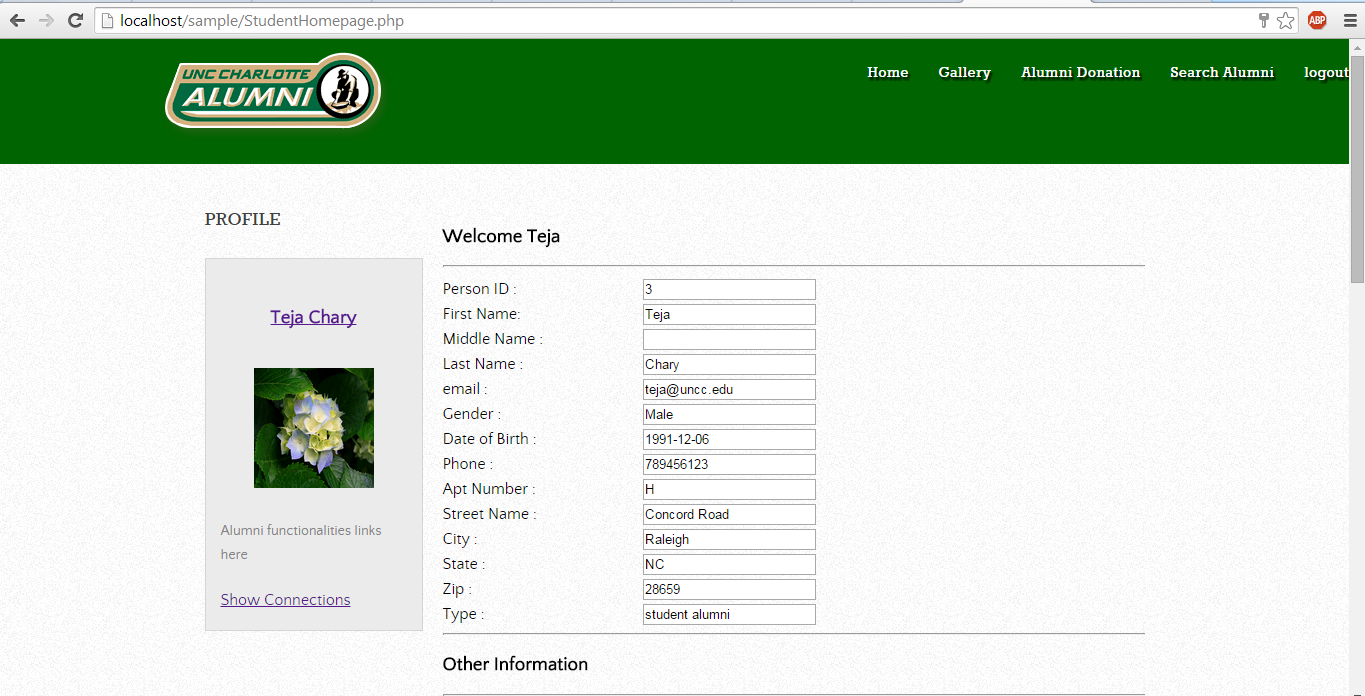
**INSERT INTO `university\_alumni`.`student\_alumni` (`person\_id`, `grad\_year`, `major\_advisor`, `major\_dept`, `awards`) VALUES ('800866867', '2011', 'Pamela Thomson', 'Information Technology', NULL);**

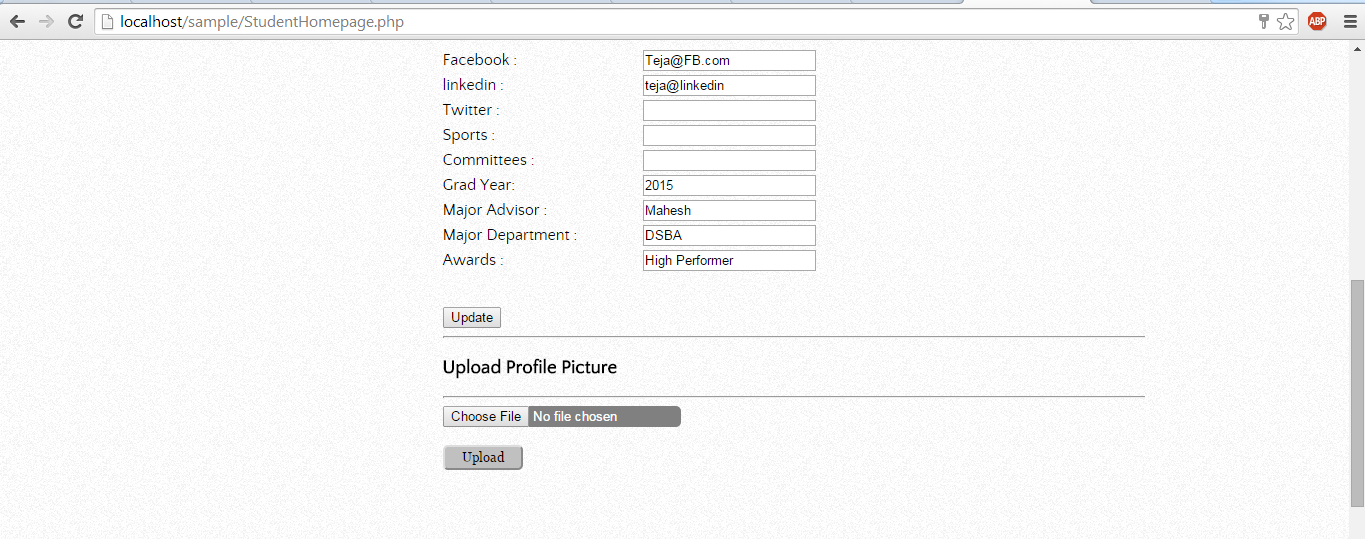
**Business Functions implemented:**

1. Create Account – To create an alumni account, the user has to fill in his details and send it to the administrator for validation.

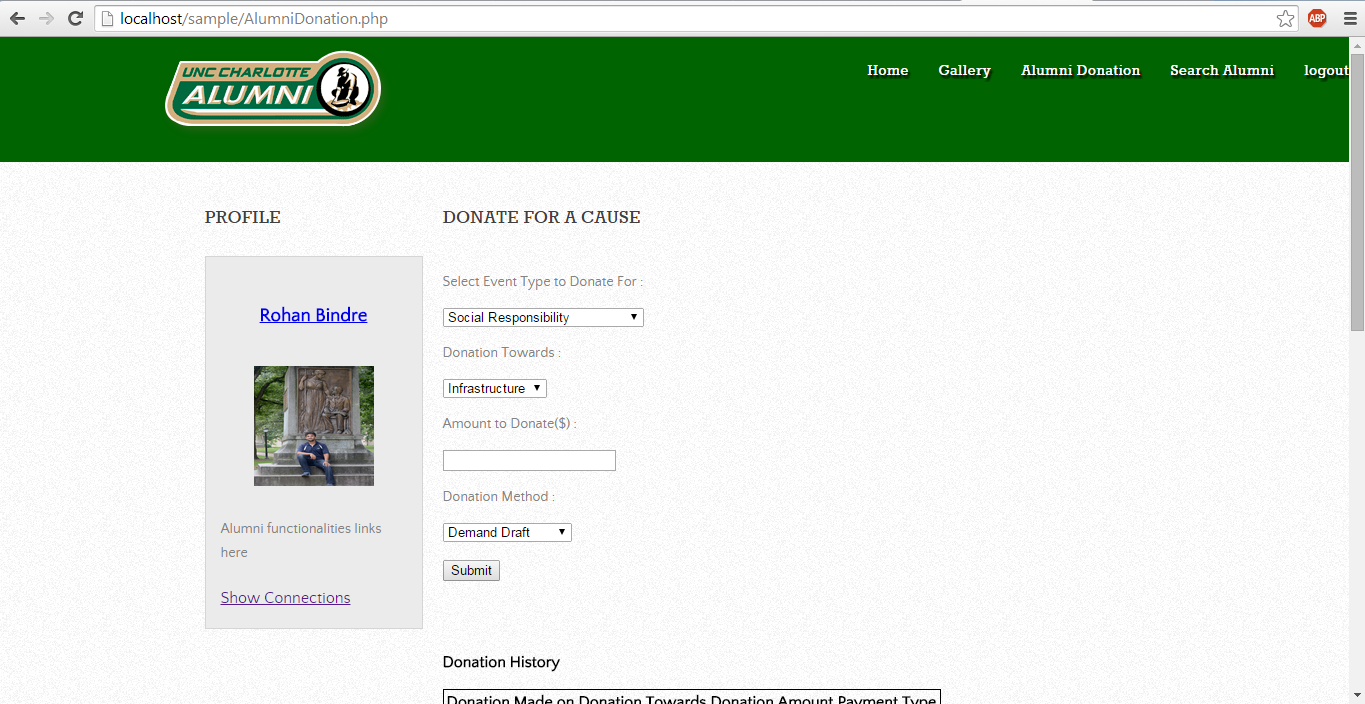


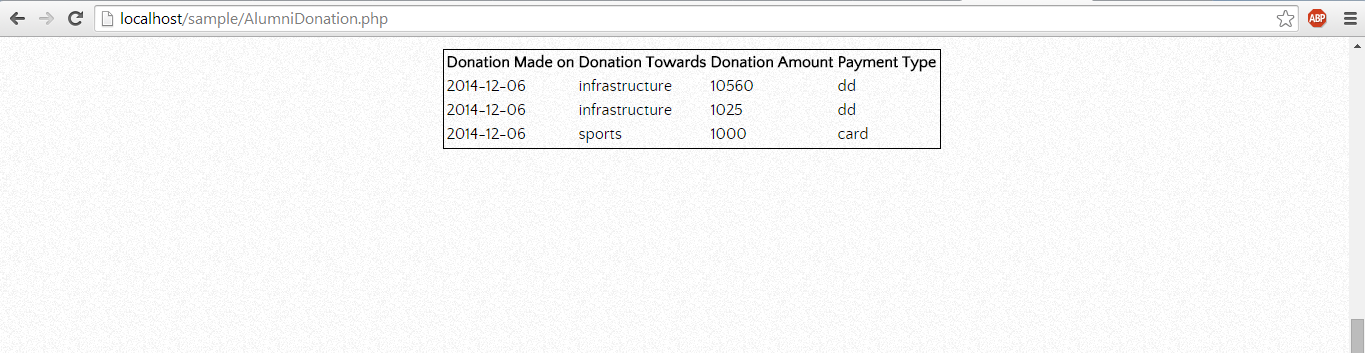




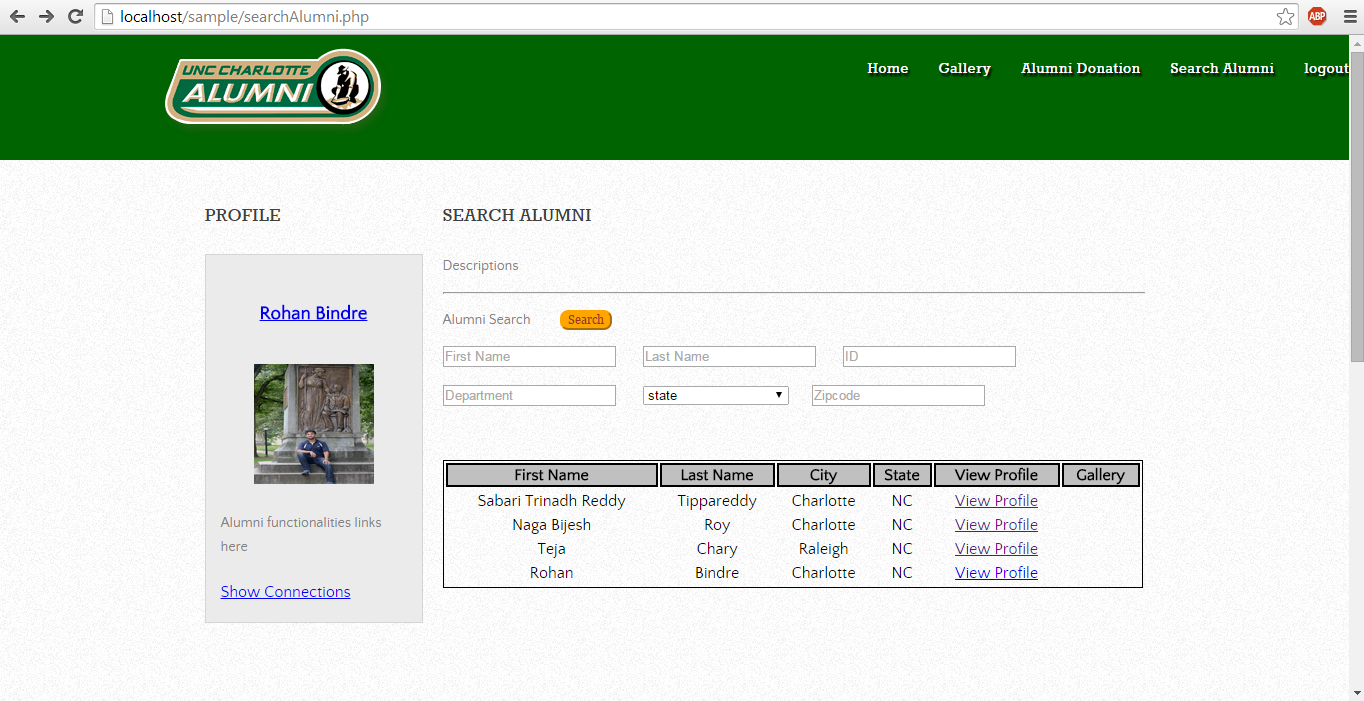


1. Donation to university – An alumni can donate to current or future events that will be displayed to him. He can donate towards infrastructure, transport, food, etc.

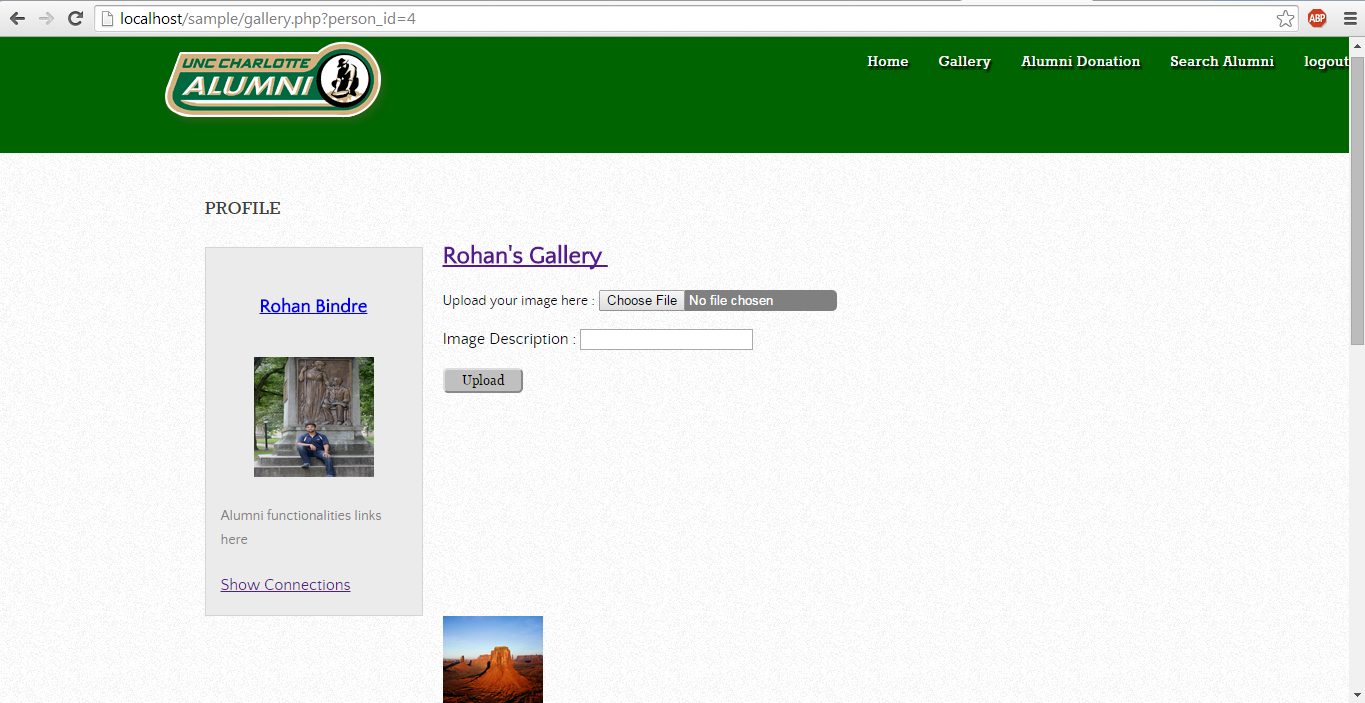


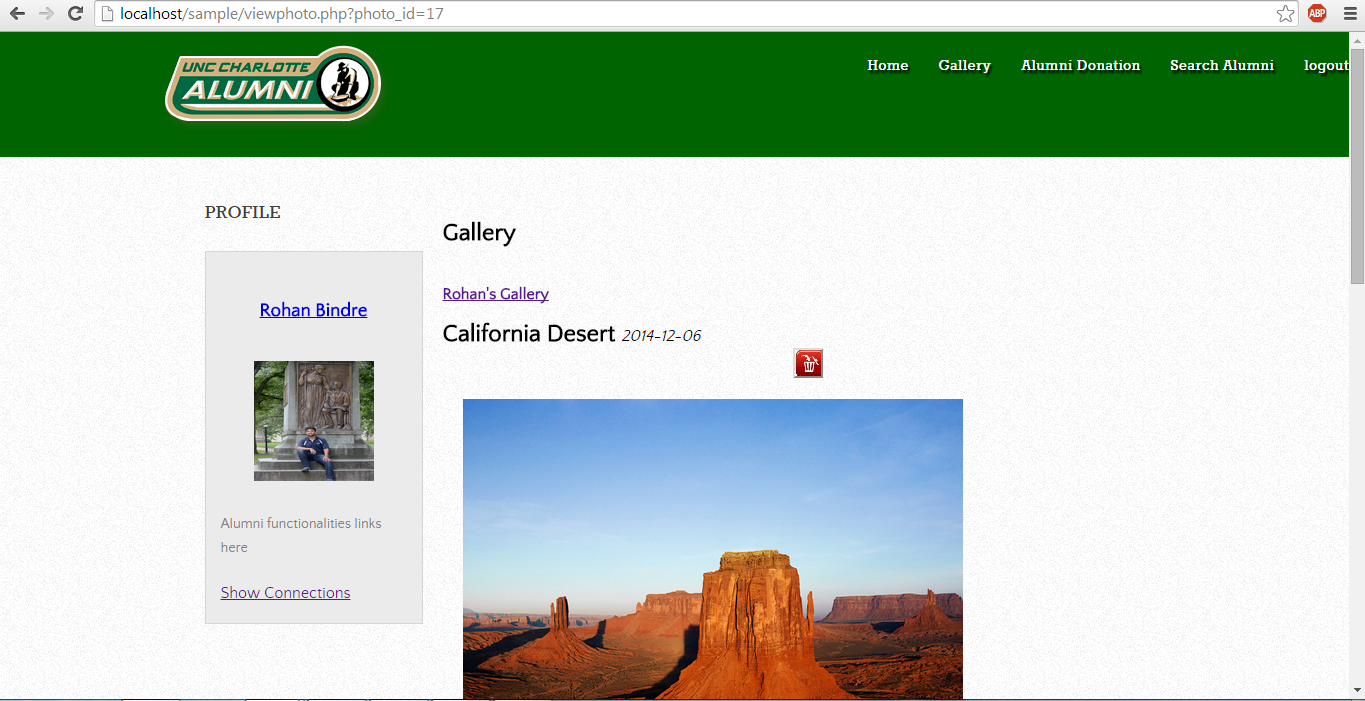


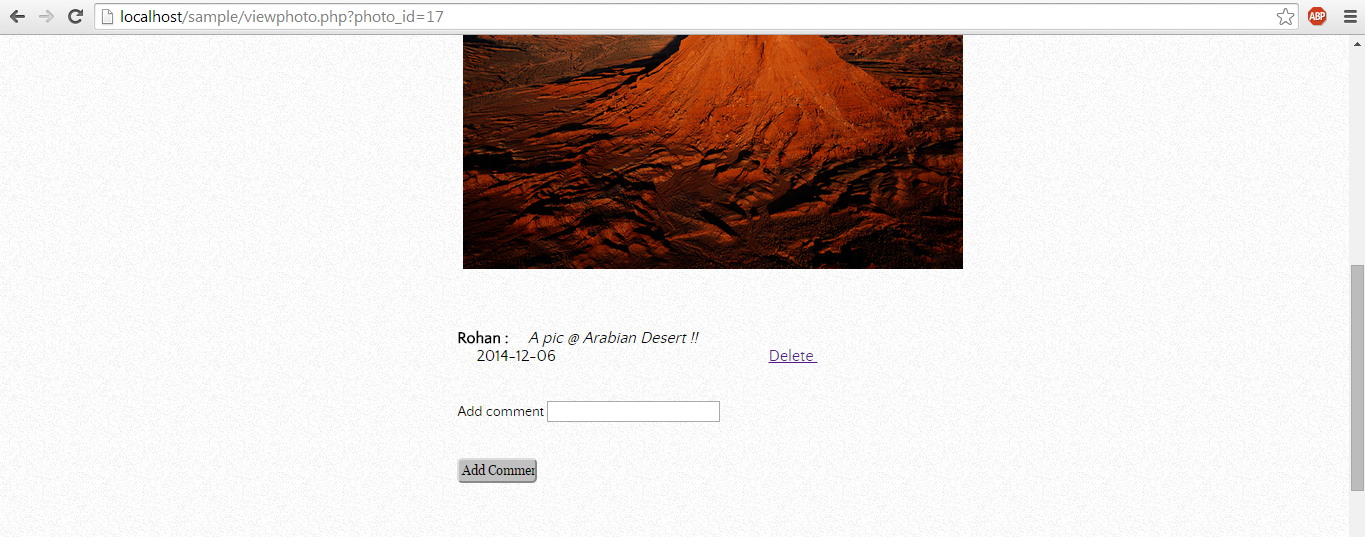
1. Search alumni – A person can search on first name, last name and geography.



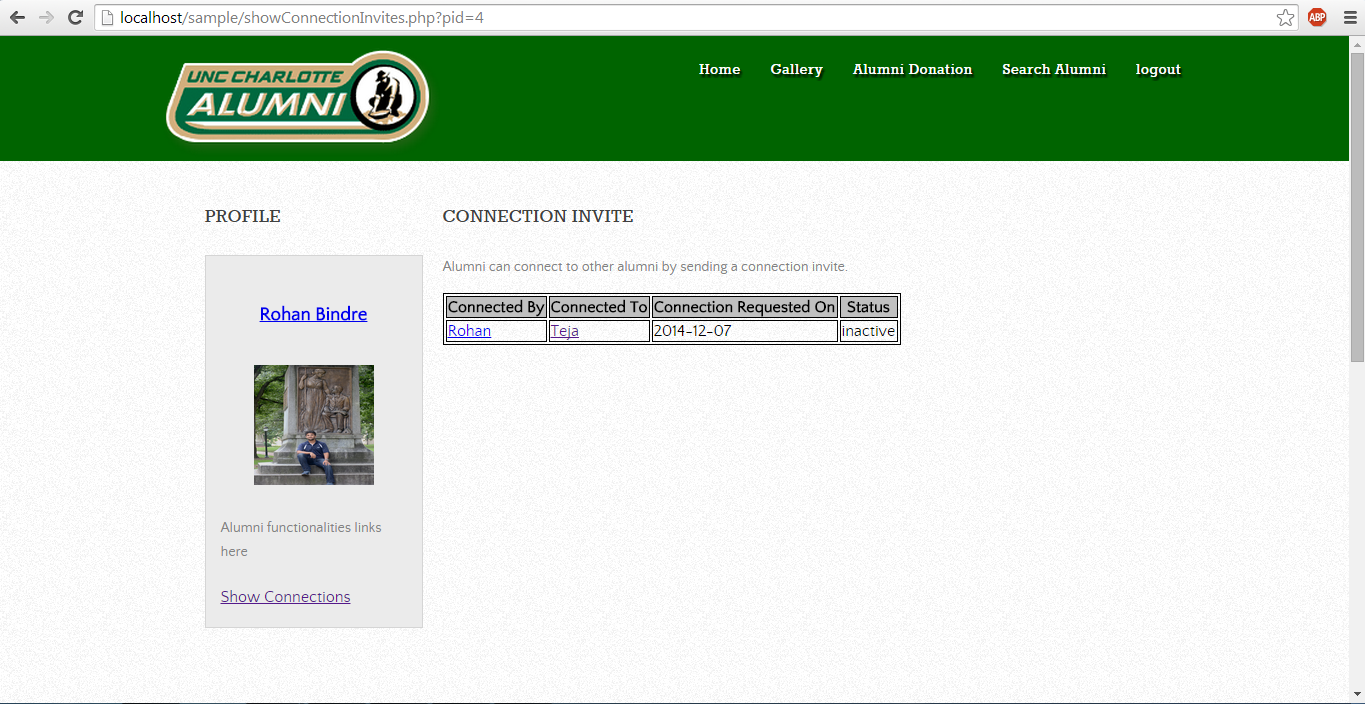
1. Upload/View Pictures to Gallery – Alumni can add photo to gallery. Connections of alumni can view these photos. They can also comment on photos.



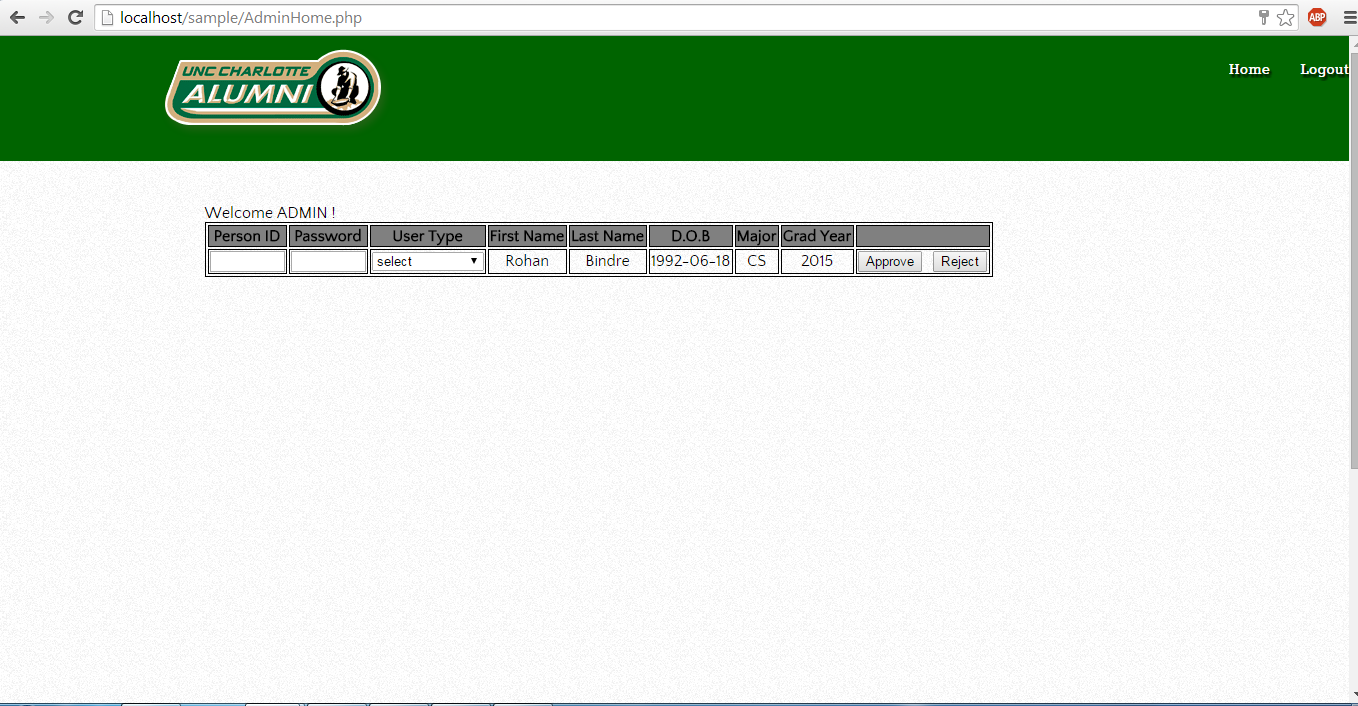




1. Send connection invite – Alumni can send connection invites to other alumni.



1. Authentication of alumni – The administrator will validate the credentials of user who wants to register as alumni. If he is a valid user, then only he can register.



**Future Scope:**

1. Creating events
2. Managing connections
3. Messaging system for alumni to communicate with each other