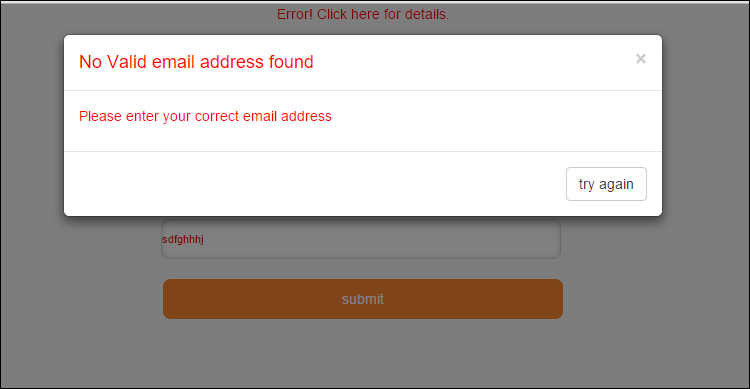
**Course-COMP453 Database Progamming**

**Team 5- Pallavi agarwal, Smruti Tatavarthy, Madhura joshi**

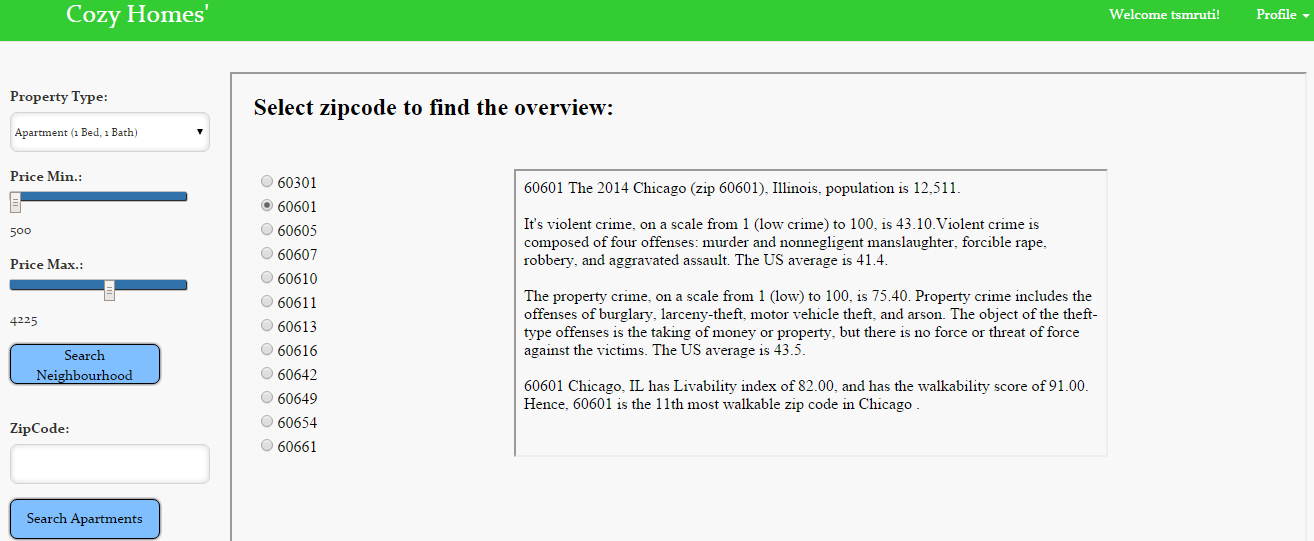
**Short Report showing fulfillment of requirements.**

1. The project is written using PHP and MySql.
2. **Highlights of the Project**

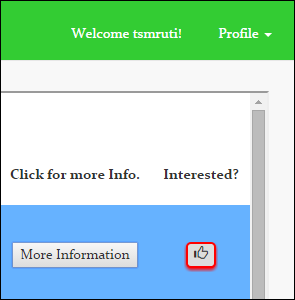
* The project has styling in the form of CSS, JavaScript, Jquery for the forms.
* Use of ***Bootstrap model*** for displaying the error messages during validation of forms.



* Use of ***Frames*** for showing the search results. When user specifies search criteria in the left side of the page, the results are shown on the right side. This gives the user better visualization of the results and the search criteria.



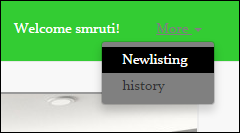
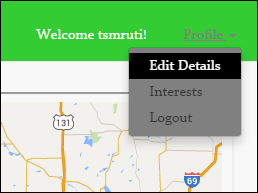
* Use of a ***like*** button---If the user likes a particular property he can hit the like button and the button turns red indicating that the property has been liked.



* Use of ***Unlike*** Button---if the user changes his mind about a property he can hit the unlike button and the property will be removed from the database



* Use of ***settings*** functionality for user/Admin—This dropdown enables users to edit their profile details, see their liked properties, history off different updates they have done to their properties.



1. **Structural change**-The DDL is altered in a way such that if any property is deleted from the database, the corresponding apartments which were present in that property will also be deleted from the database. For this we have used ***ON Delete Cascade*** by altering the apartment table.
2. **Insert SQL**- 
3. **Insert during execution of application---** This Requirement is fulfilled when a user likes a particular property, that property is inserted into the userinterest tables.

SQL=INSERT INTO userinterest SET

username = :username,

ApartmentID = :ApartmentID,

PropertyName = :PropertyName,

TypeName = :TypeName,

Price = :Price,

LeasePeriod = :LeasePeriod,

Address = :Address,

ZipCode = :ZipCode,

PhoneNo = :PhoneNo

1. **Delete-** This requirement is fulfilled when the admin wishes to delete a particular property.

SQL=DELETE FROM apartment WHERE ApartmentID = '$aptID' 

1. **Simple SQL-** This requirement is fulfilled in a lot of places. One of them is selecting type from property type. This is done for populating the drop down of the property type field.

SQL=SELECT TypeName FROM propertytype

1. **Aggregate Function-**This requirement is fulfilled on the initial admin page. The stats on the admin page show the average price of properties across the zip codes. The underlying SQL is an aggregate query.

SQL=SELECT propertytype.TypeName as TypeName, AVG(Price) as Average

FROM propertytype, apartment

WHERE apartment.TypeID=propertytype.TypeID

GROUP By apartment.TypeID";



1. **JOIN-** This requirement is fulfilled when the user gives the search criteria by specifying the different fields like zip code, property type, and price.

SQL=SELECT property.PropertyID, Apartment.ApartmentID, property.PropertyName, propertytype.TypeName, apartment.Price, apartment.LeasePeriod, property.Address property.ZipCode, property.PhoneNo, property.Rating

FROM Property, Apartment, propertytype

WHERE propertytype.TypeID= apartment.TypeID AND

apartment.PropertyID = property.PropertyID AND

propertytype.TypeName = :TypeName AND

property.ZipCode = :ZipCode AND

Price BETWEEN :minPrice AND :maxPrice';

1. **Correlated SubQuery-**This requirement is fulfilled when the user specifies the search criteria and is presented with the list of zip codes. The user can then select the required zip code for further search

SQL=SELECT DISTINCT(ZipCode) as zipcode

from property

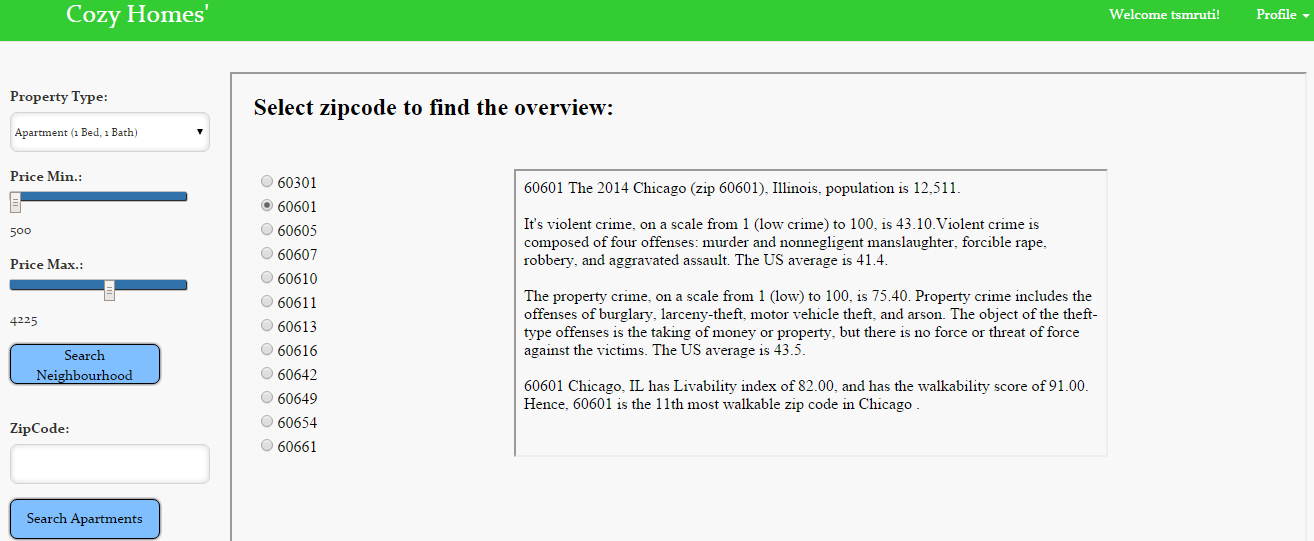
WHERE EXISTS

(SELECT \* FROM apartment, propertytype WHERE property.PropertyID = apartment.PropertyID AND propertytype.TypeID = apartment.TypeID AND

propertytype.TypeName='$type' AND

apartment.Price BETWEEN '$priceMin‘ AND'$priceMax')

ORDER BY ZipCode";



1. **SubQuery-** This requirement is fulfilled when the user clicks on his interests in the profile settings tab to see the properties that he liked.

SQL=SELECT \*

from userinterest

where UserName = :username and userinterest.ApartmentID

IN (SELECT Apartment.ApartmentID

FROM Property, Apartment, propertytype

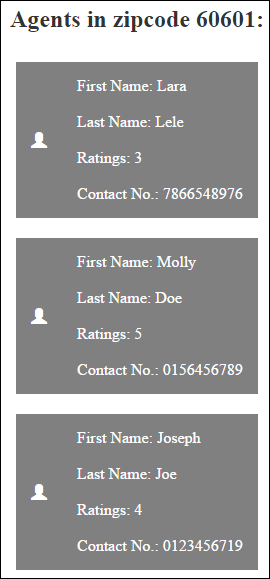
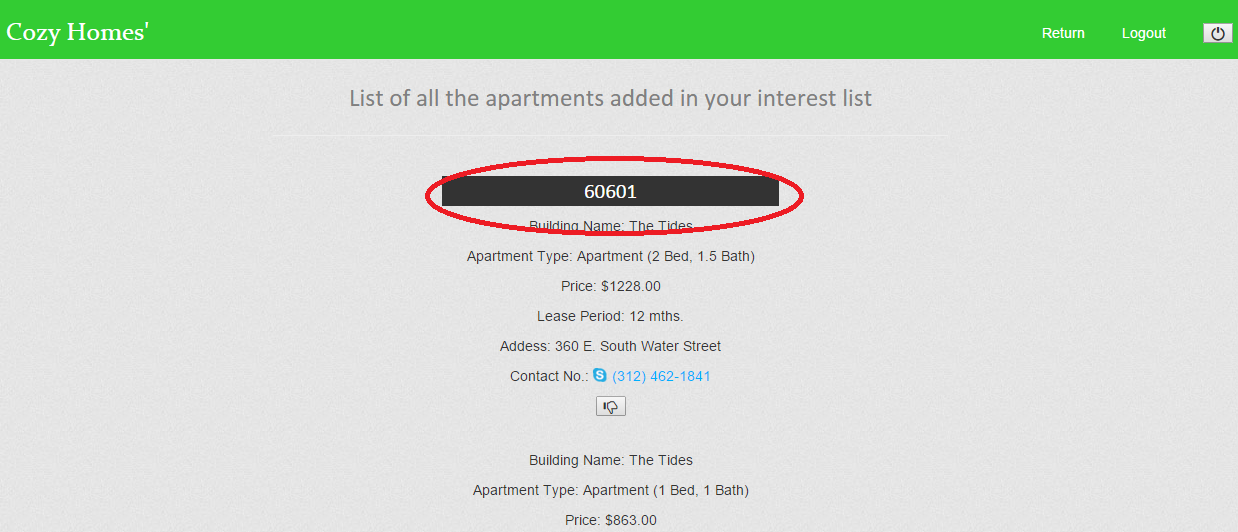
WHERE propertytype.TypeID= apartment.TypeID AND

apartment.PropertyID = property.PropertyID AND

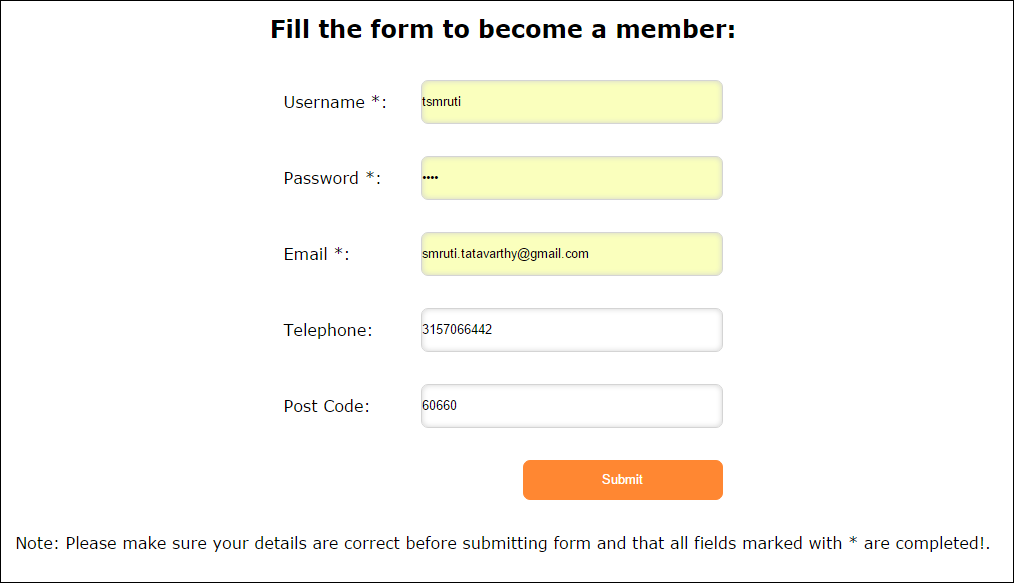
propertytype.TypeName = :TypeName AND

property.ZipCode = :ZipCode AND

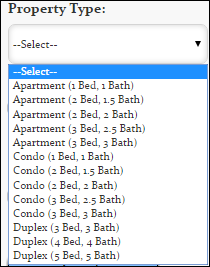
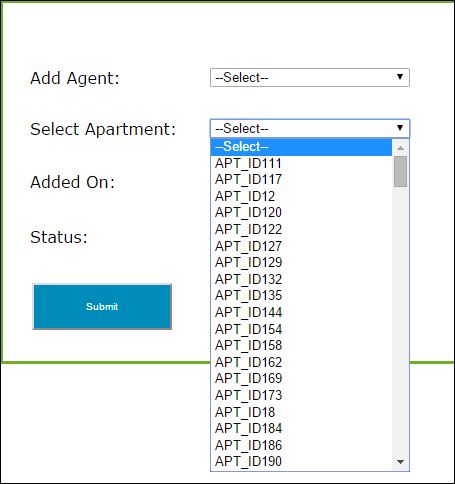
Price BETWEEN :minPrice AND :maxPrice)

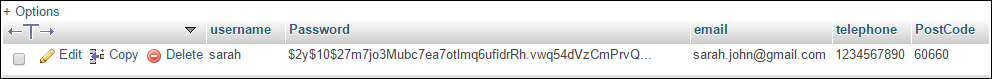


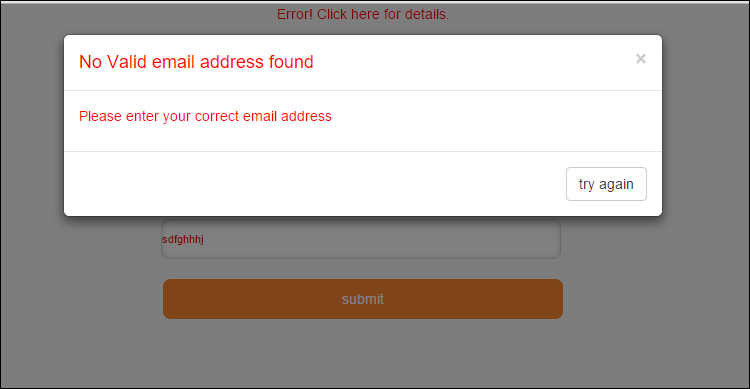
1. **FORM-**This requirement is fulfilled during the new user registration. The user needs to fill out his details and the post function will pass these variables to the listing page.



1. **GET-** This requirement is fulfilled during validation of the admin login page. After the form is validated for empty fields and a valid email address, the variables are passed to the next page and a GET is used to get the variables.
2. **Populate a field on a form or table from the database**- This requirement is fulfilled at 2 places. One is the drop down that is populated from the database for the property type and the other requirement is fulfilled in the listing page where agents dropdown is populated from the database.

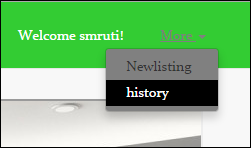
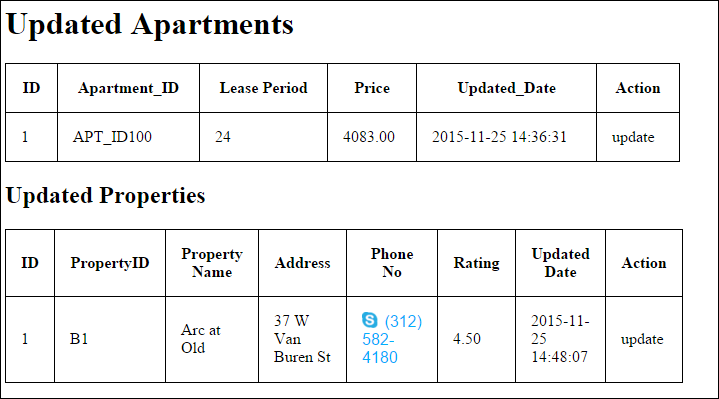
 

1. **Login and password.  Extra credit if you use bcrypt**- This requirement is fulfilled in the new user registration form. The password entered by the user is hashed and stored in the database.
2. **Check for empty data fields-** This requirement is fulfilled in login for new user, existing user, admin where the empty data is checked for and error message is displayed.

****

1. **Implement Trigger**- This requirement is fulfilled when the admin updates a particular property, the trigger will executed and the old records will be inserted in the admin update history table.

This requirement is fulfilled when the admin updates apartment details, the trigger will executed and the old records will be inserted in the admin update history table.

1. **Referential Integrity**- Property ID which is PK in ‘Property’ is a FK in ‘Apartment’ and ‘PropertyUtility’ tables. Hence when an administrator deletes a property record. The corresponding record should be deleted from both ‘Apartment’ and ‘PropertyUtility’ tables.

We have implemented this by using On Delete Cascade.

**For Property Utility**: Alter table propertyutility drop FOREIGN KEY propertyutility\_FK1;  
alter table propertyutility add CONSTRAINT propertyutility\_FK1 FOREIGN KEY(PropertyID) REFERENCES Property(PropertyID) on DELETE CASCADE;

**For Apartment**: Alter table apartment drop FOREIGN KEY Apartment\_FK2;  
alter table apartment add CONSTRAINT Apartment\_FK2 FOREIGN KEY(PropertyID) REFERENCES Property(PropertyID)  
on DELETE CASCADE;

1. **Transaction**: This requirement is fulfilled when the admin tries to add a new apartment and the price entered is negative. Without the ‘Rollback’ statement, record will be added to the database. Once the Rollback’ statement, is set the record will be rolled back from the database and pop-up will be shown.

