**DATABASE MANAGEMENT SYSTEMS**

**PROJECT**

**NAME:-SIMRAN KOUL**

**17BCE2210**

**L5+L6**

**D2 SLOT**

**SCREENSHOTS FOR OUR WEBSITE**

**GROUP MEMBERS:-**

**RAKSHIT MALHOTRA-17BCE0941**

**AGRIM NAUTIYAL-17BCE2046**

**ROHIT CHORGHE-17BCE2264**

**SIMRAN-KOUL-17BCE2210**

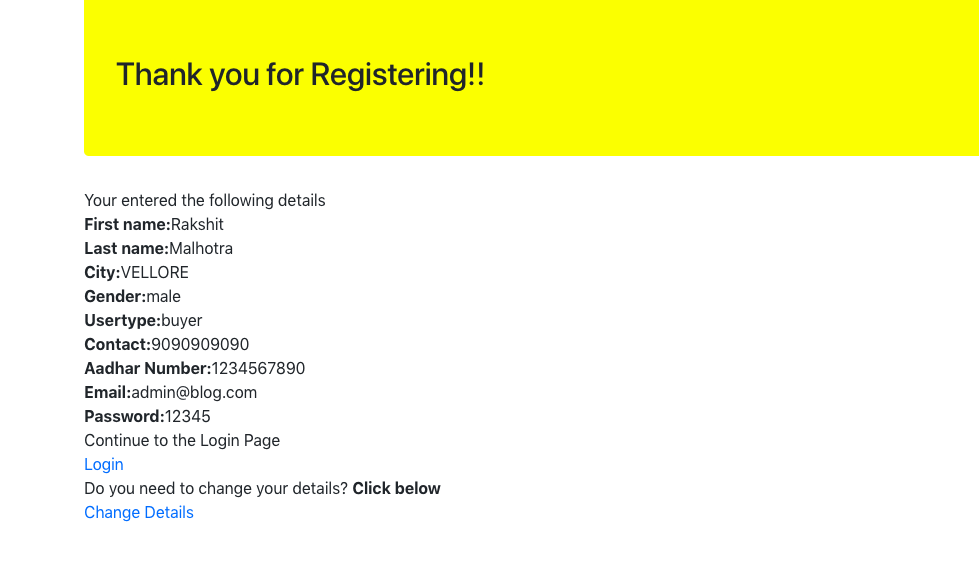
**HOME PAGE:-**



**REGISTRATION PAGE:-**



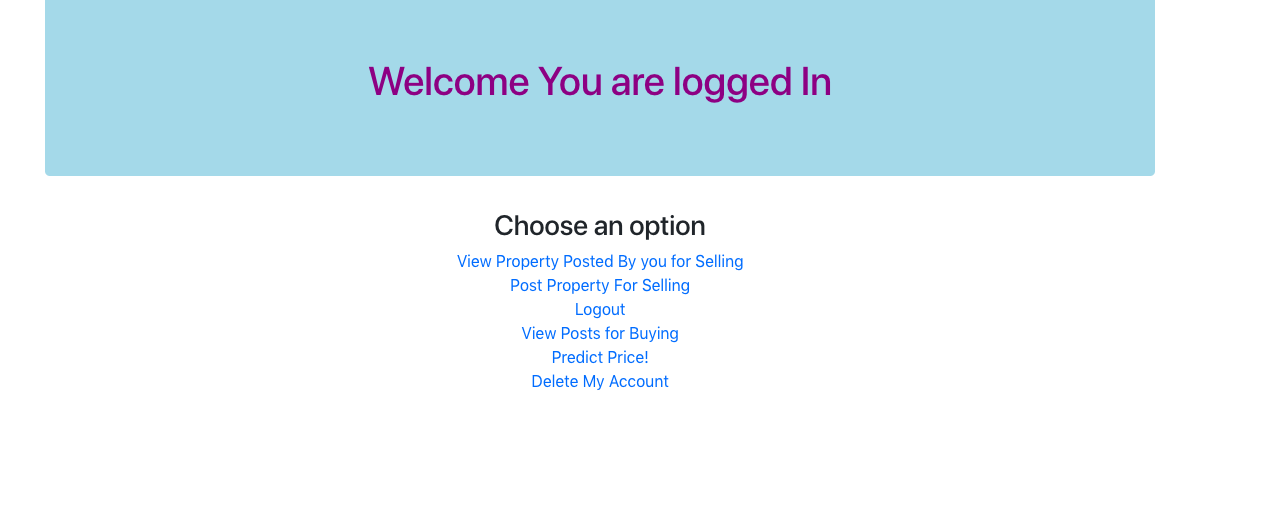
**CONFIRMATION PAGE:-**



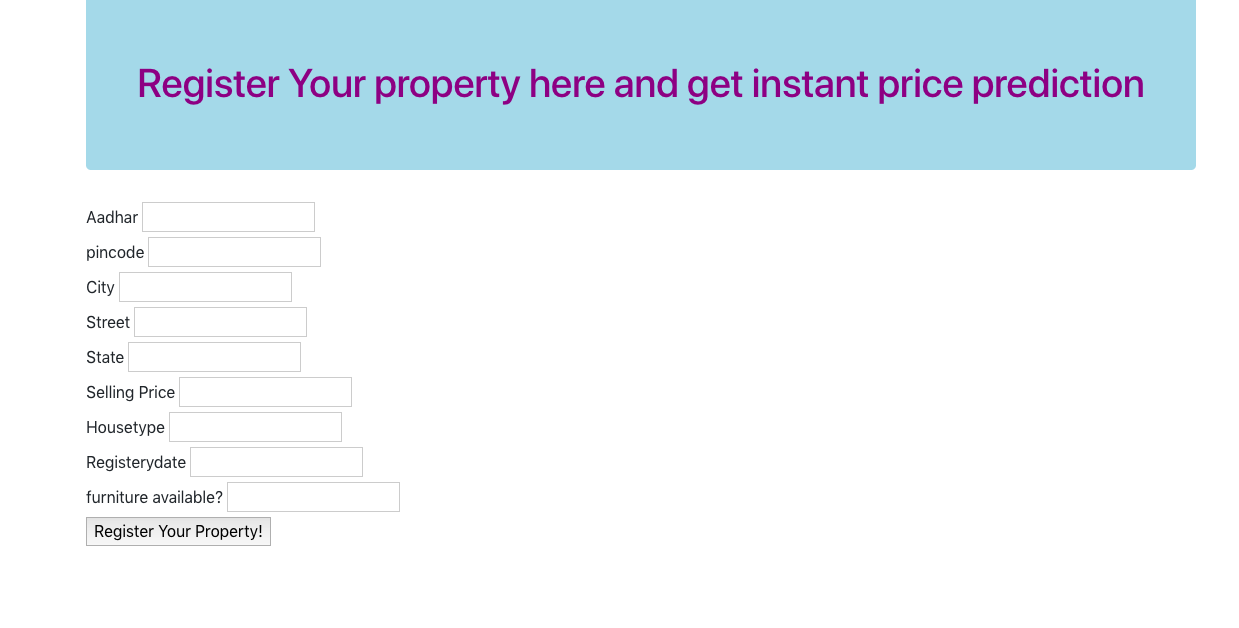
**LOGIN PAGE:-**



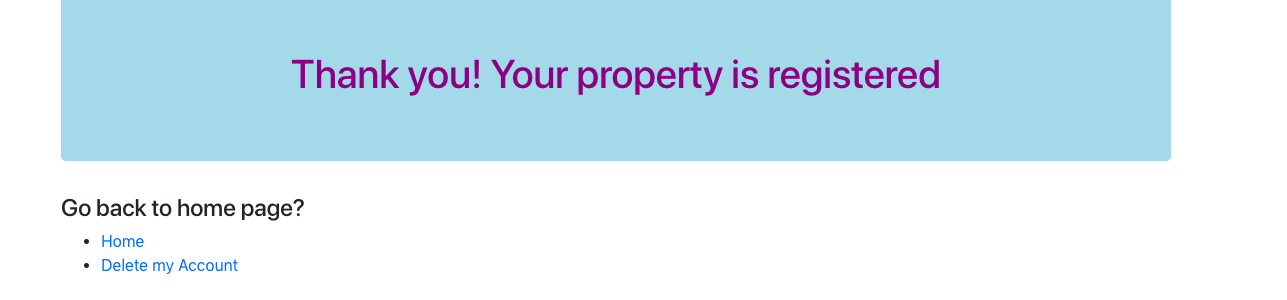
**LOGGED IN PAGE:-**



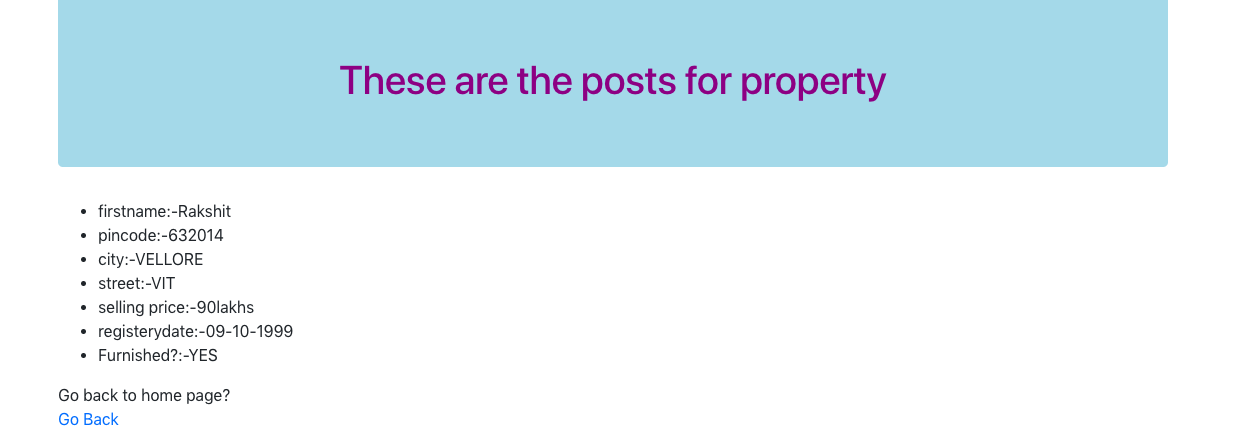
**REGISTERING THE PROPERTY:-**



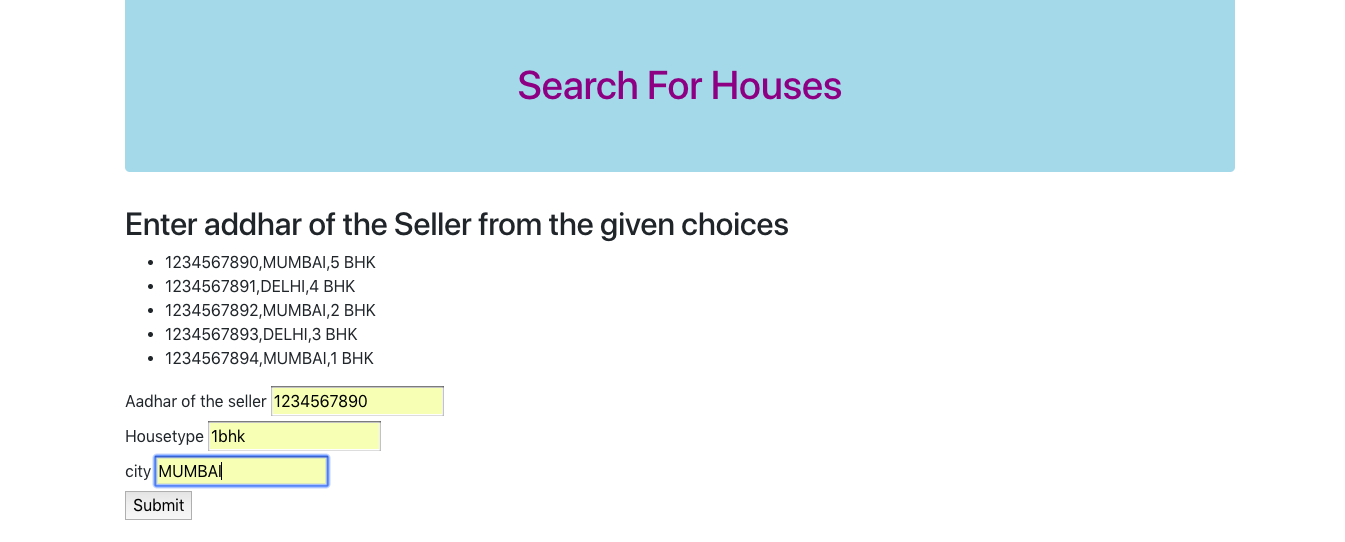
**PAGE AFTER REGISTERING:-**



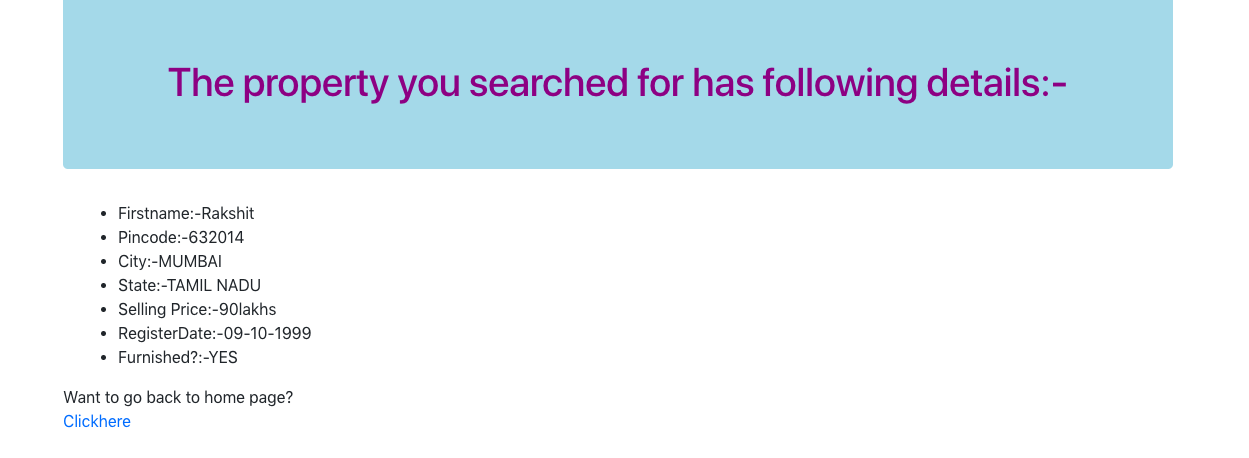
**VIEWING THE REGISTERED PROPERTY:-**



**SEARCHING FOR HOUSES:-**



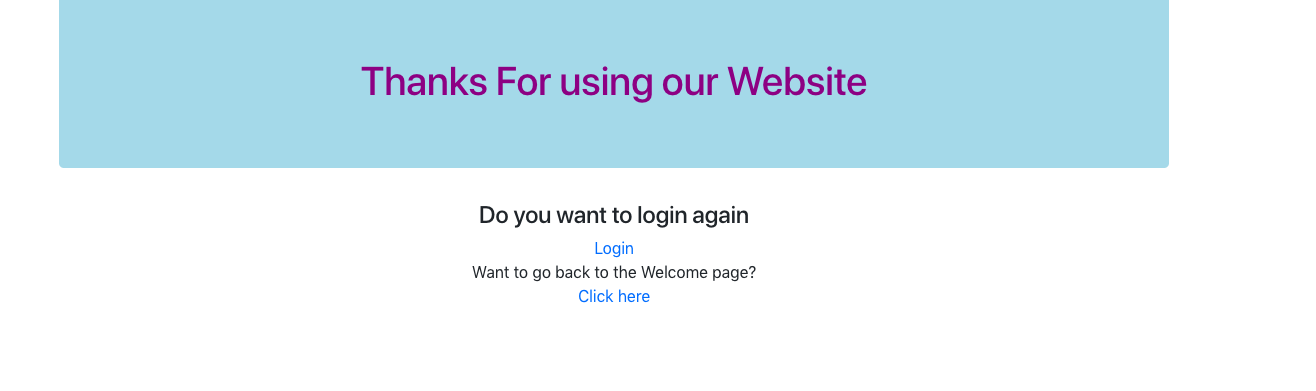
**PROPERTY SEARCHED:-**



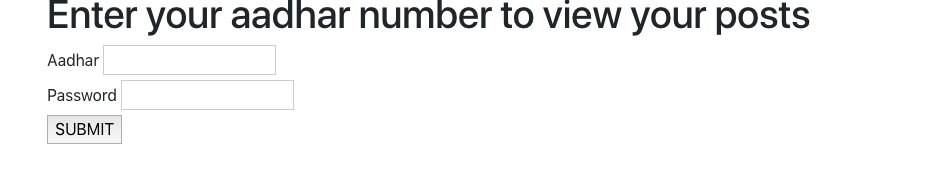
**PREDICTING PRICE FOR PROPERTY:-**



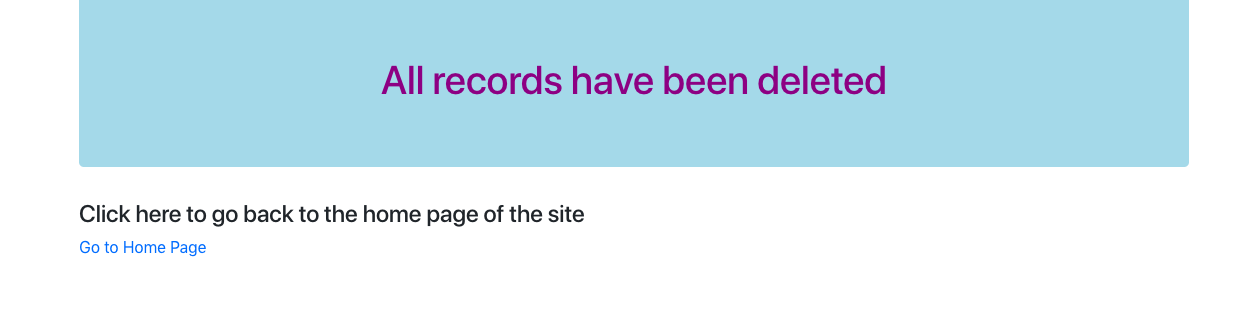
**LOGOUT:-**



**CONFIRMATION PAGE FOR DELETION OF ACCOUNT:-**



**CONFIRMATION PAGE FOR DELETION OF ACCOUNT:-**



**Characteristics of our nosql database (Redis)**

**Redis**

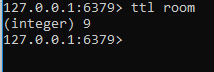
Redis has a huge advantage over the traditional method database management method. Redis belongs to what we call as a nosql database. The word **nosql** means **“Not Only Sql”** which means that the data is stored in a format different to the tabular method in sql. The data in redis is stored in a format of **key-value** pair which means that a key is assigned to a data value and the data is referred by the key similar to a linked list in data structures . An example can be shown as:



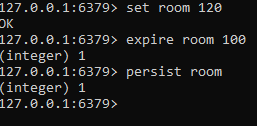
In this example a key room has been associated with a value 100. Furthermore, the data 100 is referred by the key ***room.*** Another observation made here is the fact that I didn’t have to mention the data type for the number 100. The system takes it on it’s own. This is another advantage over the sql database. We can even set the time limit for the key-value pair to expire if it’s use is temporary. For ex:



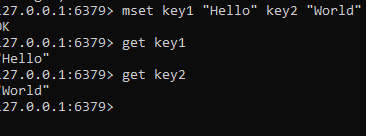
Here 100 represents the time left (in s) for the key **room** to expire. We can check the status of the key as:



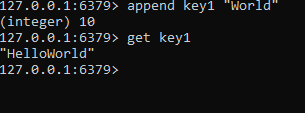
Here 9 represents time (in s) left for the key **room** to expire. If you don’t want to delete the key in the middle of it’s expiry then:



If you want to name more than one keys at the same time then:



If you want to add data to the value of a certain key then:



If you want to rename a key then:

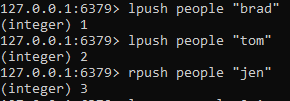


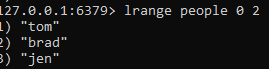
**The various data types in redis are:**

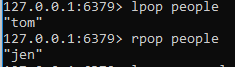
1. **Lists**
2. **Sets**
3. **Sorted Sets**
4. **Hashes**

**Lists**

Lists are basically data types which are similar to an array. The difference between a list and the other data types in redis is that the addition and deletion of data in a list is bi-directional. We can add/delete data from the right as well as from the left. For ex:

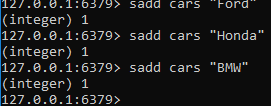






**Sets**

Sets is a form of a data type similar to key-value pair. It’s general form can be shown as:



Here **cars**  represents the field and **Ford,Honda,BMW**  are the data values associated with **cars**

**Sorted Sets**

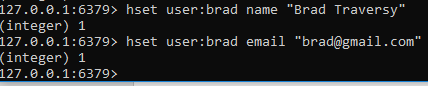
Sorted sets are similar to sets but the difference in sorted sets is that each value is associated with a score. This score helps in giving priority to a particular record. For ex:



Here **zadd** is the command, **users** is the field, **(1,2,3)**  are the scores and (**Brad, John and Mike)** are the values.

**Hashes**

Hashes are the most commonly used data types in redis. This is mainly because each record of a hash has a hash-key which uniquely determines a record and acts more so as a primary key in the case of sql database. The data format is a shown:



In this example **user:brad** is a hash-key, **(name, email)** are field names and the last entities are values.