

National Institute of Technology – Karnataka, Surathkal



Database Management Systems Lab Project

On

Real Estate Management Systems **REALTOR**

By

Mohit Bhasi – 16CO126

Prince Abhinav – 16CO134

ABSTRACT

Realtor is developed to build a platform that serves as a link between home seekers, real estate developers and land owners. It retrieves the information about the supply, demand, and discovery of real estate with a click of a button.

The main aim of this project is to make a connection between home seekers, real estate developers and land owners. This project intends to include various features such as the home seekers can directly search for the home which they want to buy, they can also directly connect to landlord if they want to buy their own property by giving certain specification about the property which they want to buy. Similarly, the builder can buy lands from landlord. There is a login page where the landlord, the builder can login and list their properties for sale. Once this is done the buyer can login and see all the properties which are available. Landlords and builders are expected to keep updating their properties on our website. The main aim of this project is to apply the knowledge the of MySQL taught under DBMS Subject in a real-life project

This software package has been developed using the powerful coding tools of HTML, CSS, Bootstrap at the Front End and Django, MySQL Server at the Back End. The software is very user friendly. This version of the software has multi-user approach. For further enhancement or development of the package, user's feedback will be considered.

CONTENTS

1. Introduction	
1.1 Purpose	
1.2 Objective	
2. Requirement specifications	
2.1 Hardware specifications	
2.2 Software specifications	
3. Database Design	
3.1 ER Diagram	
3.2 Relational Database Design	
3.3 Constraints in Relation Schema	
4. Project Components	
4.1 Front End Design	
4.2 Security Measures	
5. Results & Discussions	
6. Conclusion	

List of Tables and Figures

1. ER Model.....	09
2. Relational Schema.....	10
3. Home Buyers table.....	06
4. Project Table.....	06
5. Builder Table.....	07
6. Land Table.....	07
7. Landlord Table.....	08
8. Key Constraints Table.....	10
9. Cardinality Ratio Table.....	11
10. Home page Figure.....	12
11. Search Page Figure.....	12
12. Details Page Figure.....	13
13. Signup Page Figure.....	14
14. Login Page Figure.....	14
15. Dashboard Page Figure.....	15

Introduction

1.1 Purpose:

Any type of user can sign up on our platform and reap the benefits of the services we provide. As a landlord, the user puts the necessary details of the land they wish to sell in our website. Depending on the size of the land, either an individual or the real estate developer can connect with the land owner and go inspect the land for sale. As a real estate developer, the user can post listing of their recent housing projects which may be for sale or rent. Home seekers have a variety of actions to choose from. They can contact land owners through our platform to if they wish to purchase land. They can also contact real estate developers if they wish to buy a property or get a house on rent.

Django was chosen to design this application. Django is a Framework in which python is used as a language. Applications created with it can be used on a standalone machine as well as on distributed network. More over applications developed in Django can be extended to Internet based applications.

1.2 Objective :

In this world of growing technologies everything has been computerized. With increase in demand of people for getting a property and land as well, there is a need of a system which can maintain records of buyers of the property, builders and the landlords. With this system, people need not to go to sellers themselves and get the deal fixed. This can be done with the help of our app by simply registering and viewing as per the demand. This is also helpful for the property sellers as they can also come and register on our website and can give the description about their property, which will be viewed by the buyers.

Requirement Analysis and specifications

The aim of the system is to develop “Realtor” software, which should automate the process of making a connection between landlord, home seekers, builders. The system is supposed to be used as a web app, which will be computerized one. Therefore, the proposed system must be able to function only in the circumstance where the user will be having a laptop or a pc.

2.1 Software Requirements

- **Web Server:** The platform is going to be hosted on the web with Python as back-end and the server will be powered by Django
- **DBMS:** All the data will be stored in structured tables which will be implemented using MySQL, an open source relational database management system.
- **Development:** For development phase of our system we decided again on Django products. Our development platform will be Python and we are planning to use the following tools and languages.
 - Python, MySQL programming language for main development
 - HTML, CSS, JavaScript, MaterializeCSS, Twitter Bootstrap and Front Awesome is used to improve for aesthetics of the website. All the data will be transmitted in JSON format and will strictly follow Representation State Transfer(REST)
 - Django web services
 - Sublime Text as development tool.

Other Development Software:

- Windows 10 operating system
- MS Office for reports or any other documents.
- Lucid-Chart, erdplus for diagrams

2.2 Hardware Requirements

- **Web Server:** We need a reliable web server for our system. This machine must be fast and must show high performance in all situations. At least 512MB RAM and Pentium 4 2000MHz processor seems to be the minimum requirements for this machine. Any IBM, HP machine can be selected for this purpose.
- **Database Server:** Since our system requires a huge amount of data to be stored, we will need an extra machine that will serve as a database. At least 40GB storage capacity is needed for this system. And this machine must also be a high performance machine. An IBM machine like xseries 382 may be a suitable choice for this purpose.

Database Design

3.1. ER Diagram

This ER diagram represents the model of Realtor. The entity-relationship diagram of Realtor show all the visual instruments of database tables and relationship between home seekers, Builder , Landlords etc. It used structured data and define relationship between structured data groups of Realtor functionalities. The Relations are Buy, Contact, Owns etc.

Employee management System entities and attributes:

- Home Buyers Entity:

Attribute Name	Attribute Meaning	Attribute Value
Buyer Id	Buyer's ID	Int(11)
Address	Address of Home buyer	Varchar(20)
Contact Number	Contact number of home buyer	Int(10)
FirstName	First name of Home buyer	Varchar(20)
MiddleName	Middle name of Home buyer	Varchar(20)
LastName	Last name of Home buyer	Varchar(20)
Password	Home buyers need to set password	Varchar(20)
Email ID	Email ID of Home buyer	Varchar(20)

- Project Entity

Attribute Name	Attribute Meaning	Attribute Value
Project ID	Project's ID	Varchar(20)
Description	About the Project	Varchar(20)
Address	Address of the Project	Varchar(20)

Price	Price of the Project	Int(10)
Buyers	Buyers of the Project	Varchar(20)
Builder ID	ID of the builder who owns it	Varchar(20)
Number of Bedrooms	Number of bedrooms in the house	Int(10)

- Builder

Attribute Name	Attribute Meaning	Attribute Value
Contact Number	Contact number of Builder	Int(10)
Password	Password set by the builder	Varchar(20)
First Name	First name of builder	Varchar(20)
Middle Name	Middle name of builder	Varchar(20)
Last Name	Last name of builder	Varchar(20)
Office Address	Office address of builder	Varchar(20)
Email ID	Email ID of builder	Varchar(20)
Builder ID	Builder ID	Varchar(20)

- Land Entity

Attribute Name	Attribute Meaning	Attribute Value
Landlord ID	Landlord's ID	Varchar(20)
Address	Address of the land	Varchar(20)
Price	Price of the land	Int(20)

Size	Size of the land	Int(20)
Land ID	Land's ID	Varchar(20)
Owner ID	Owner's ID	Varchar(20)

- Landlord Entity

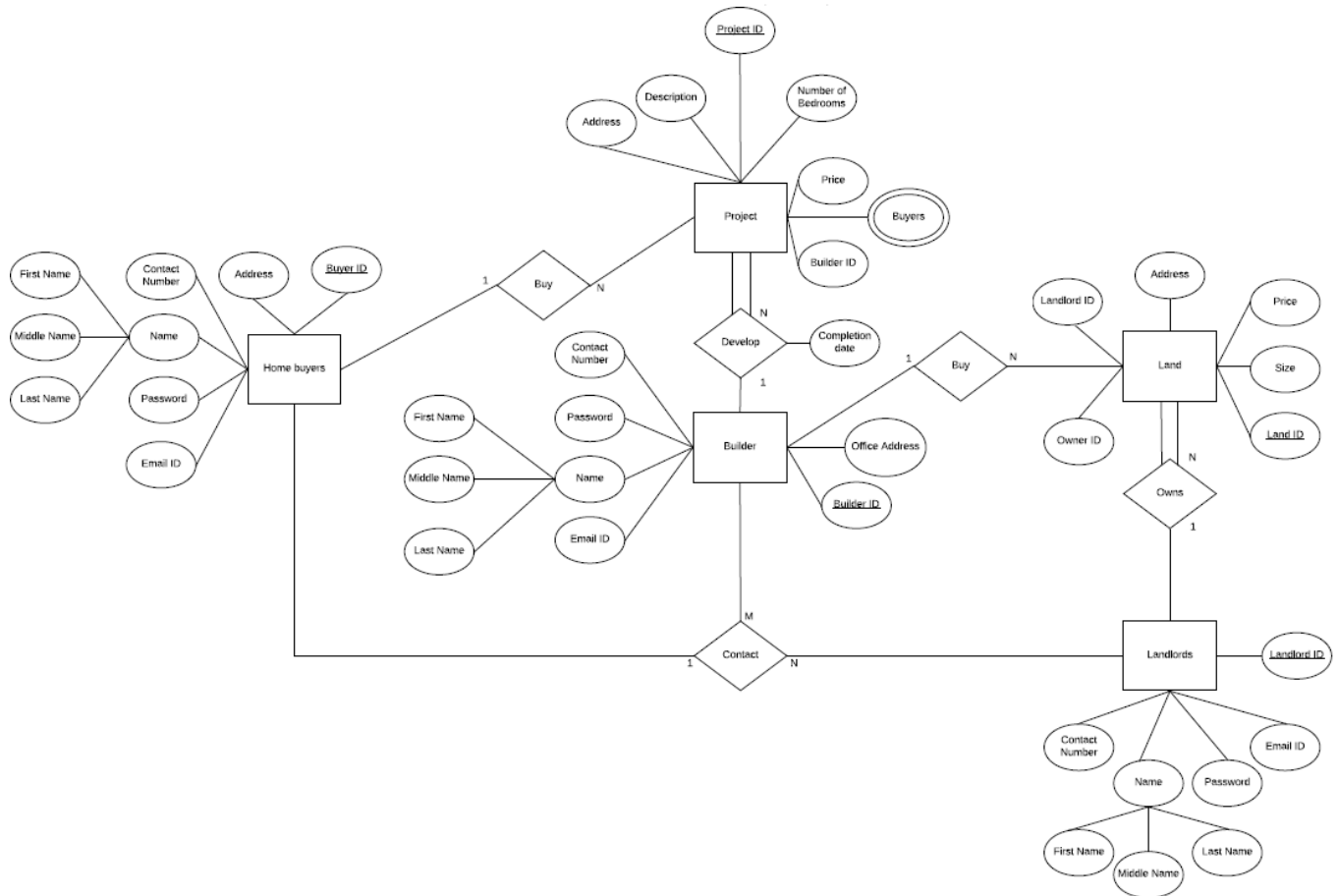
Attribute Name	Attribute Meaning	Attribute Value
Landlord ID	Landlord's ID	Varchar(20)
Email ID	Email ID of landlord	Varchar(20)
Password	Password set by the landlord	Varchar(20)
Name	Name of the Landlord	Varchar(20)
Contact Number	Contact number of the landlord	Int(10)

The entity is a concept or object in which the piece of information can be stored. There are three types of relationship between entities. They are as follows:

- **One to One(1-1):** This relationship specifies that one instance of an entity is associated with another instance of an entity.
- **One to Many(1-N):** This relationship specifies that one instance of an entity is associated with zero or many other instances of another entity.
- **Many to Many(N-N):** This relationship specifies that one instance of an entity is associated with zero or many other instances of another entity.

Residential Brokerage Platform

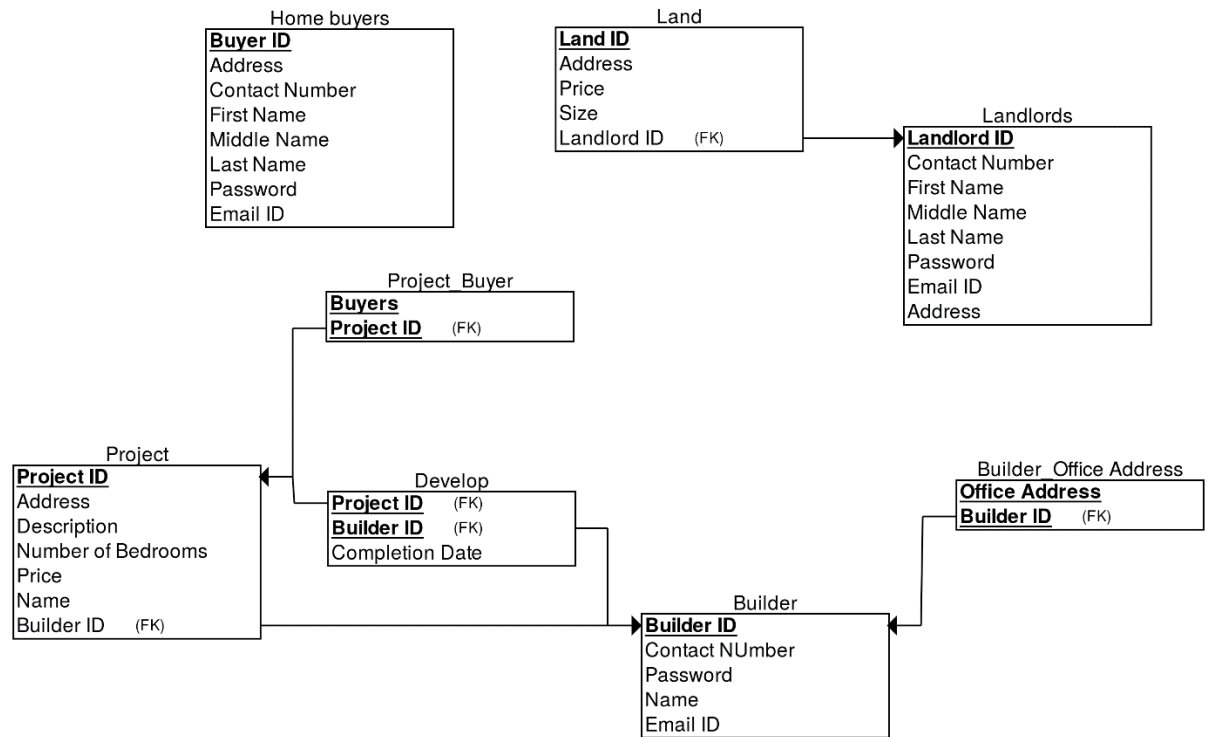
Mohit Bhasi (16CO126)
Prince Abhinav (16CO134)



3.2 Relational Database design

In our ER Diagram as the Home buyers is strong entity so it will be mapped in relational schema with all of its attributes and in this as the name is a composite attribute so in the schema name will be replaced with all of its composite attributes, next in our schema there is a Project table and as project also a strong entity but as the relation between the builder and project is one to many and also there is a full participation of project hence the project table will acquire the primary key of Builder entity as a foreign key and rest all of the project attributes will be in the project table. Develop table will have the primary key of project and builder and also its attribute as it is a type of Binary 1:1 relationship. Builder is also a strong entity hence the mapping will be having all its attributes with Builder Id as its primary key. Landlords mapping is also the mapping of strong entity so it will be having all of its attributes with Landlord ID as primary key value. In land table there will be the attribute of land entity with Land ID as the primary key along with it there will be a foreign key which will be Landlord ID as the relationship between Landlord and land is one to many with land as full participation hence land table will have the primary key of Landlord Table as foreign key.

Relational schema diagram for Realtor



3.3 Constraints in relation Schema:

Give all the types of constraints with explanations that you have used for your project. For example:

- Key Constraints

Relation	Primary Key	Foreign Key
Home Buyers	Buyer ID	
Land	Land ID	Landlord ID
Landlords	Landlord ID	
Project Buyer	Buyers	Project ID

Project	Project ID	Builder ID
Develop		Project ID, Builder ID
Builder Office Address	Office Address	Builder ID

- Cardinality Ratio

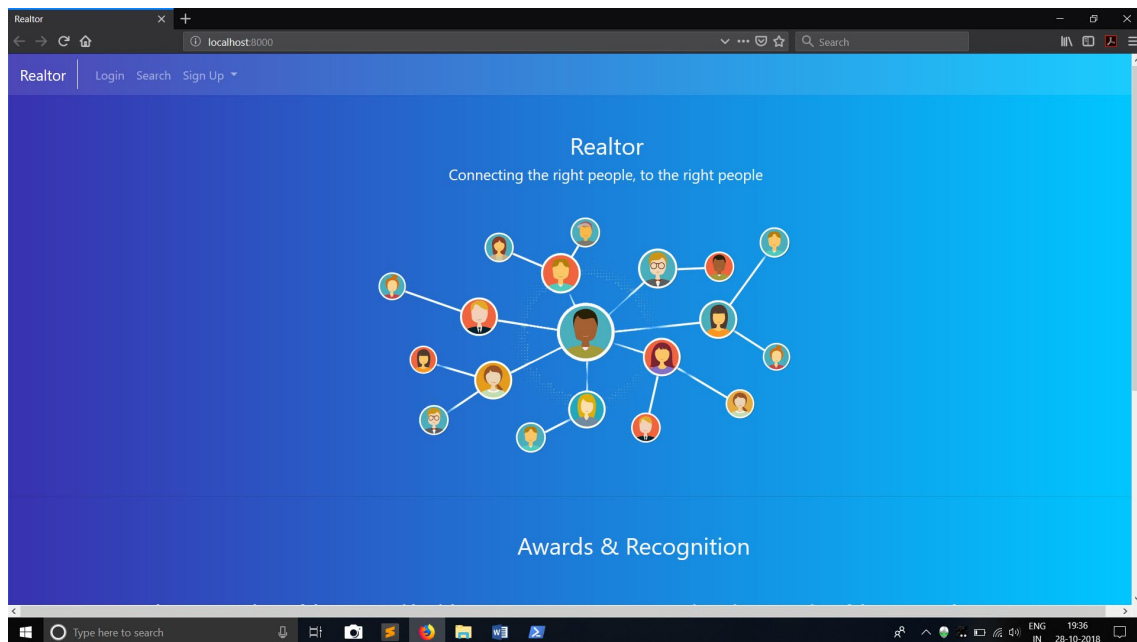
Relation	1:1	N:1	M:N
Buy	No	Yes	No
Develop	No	Yes	No
Contact	No	No	Yes
Owns	No	Yes	No

Project Components

4.1 Front End Design:

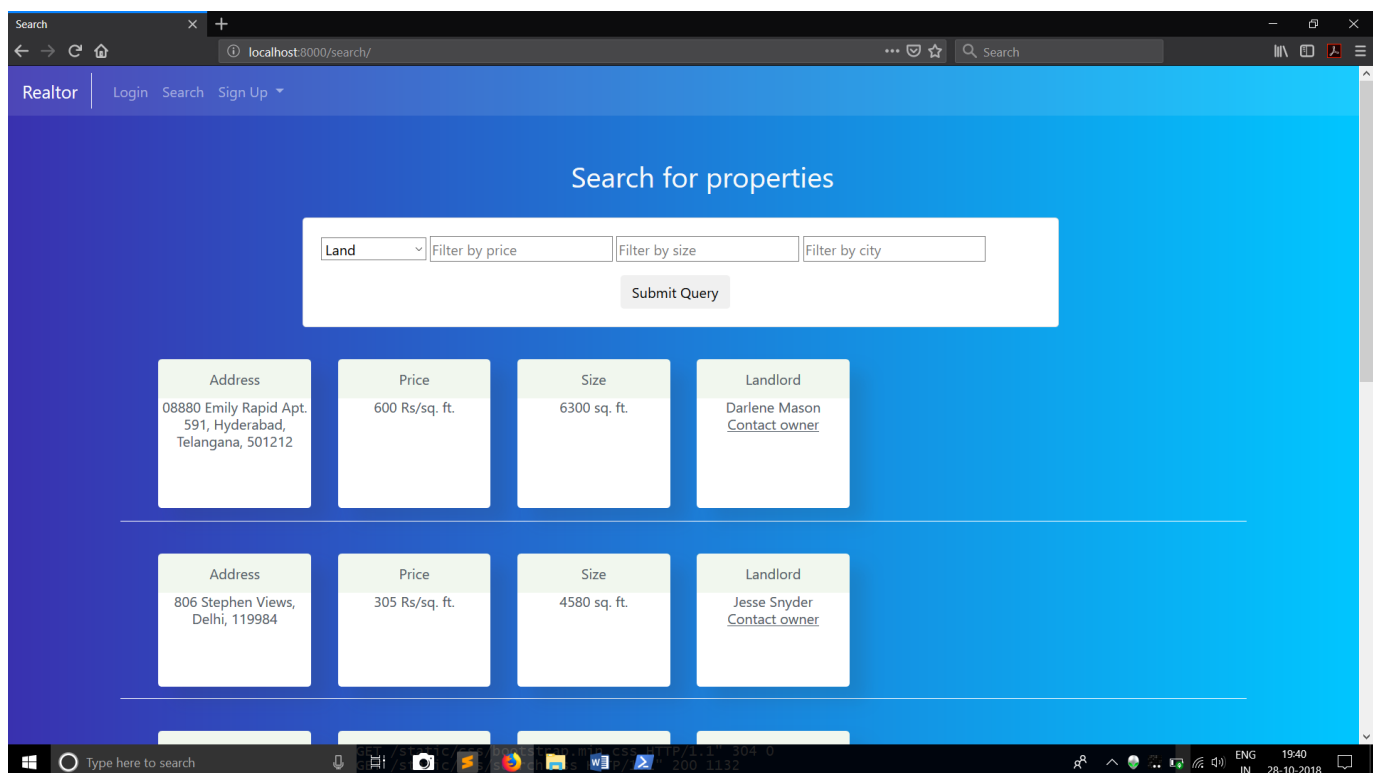
a) Home Page

The landing page of the website will be the first page the users use. The page is minimal and easy to navigate. The navigation bar helps the user perform an action of choice.



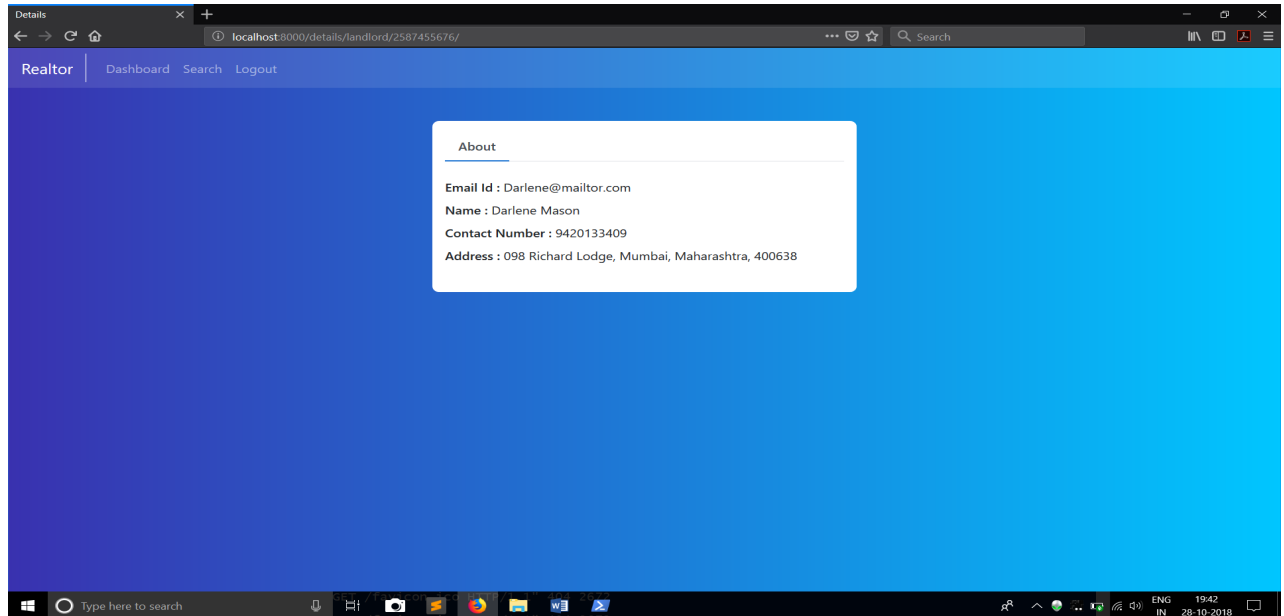
b) Search Page

The search page can be used by users to filter out properties and projects based on price, size and city.



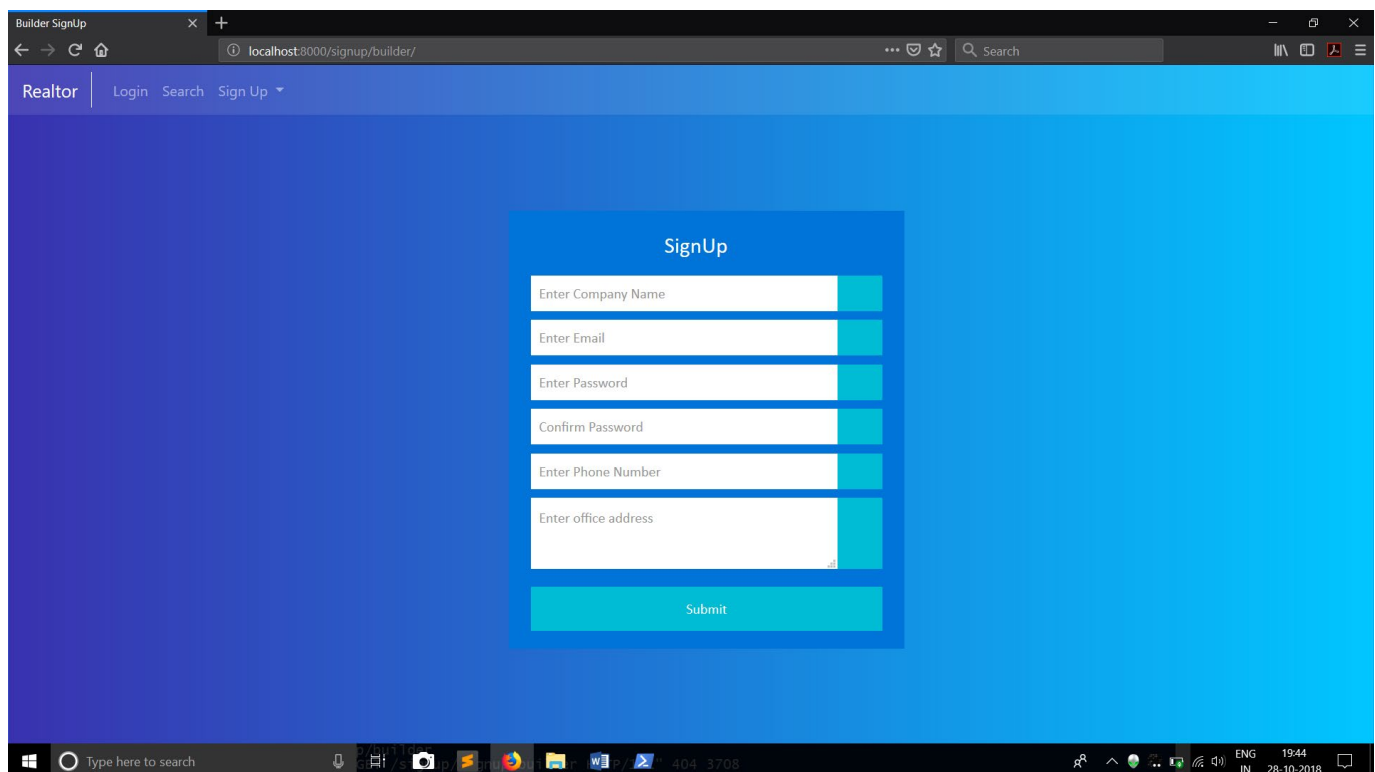
c) Details Page

A crucial page in the website, this page displays all the details of a user, be it a landlord, home buyer or a builder



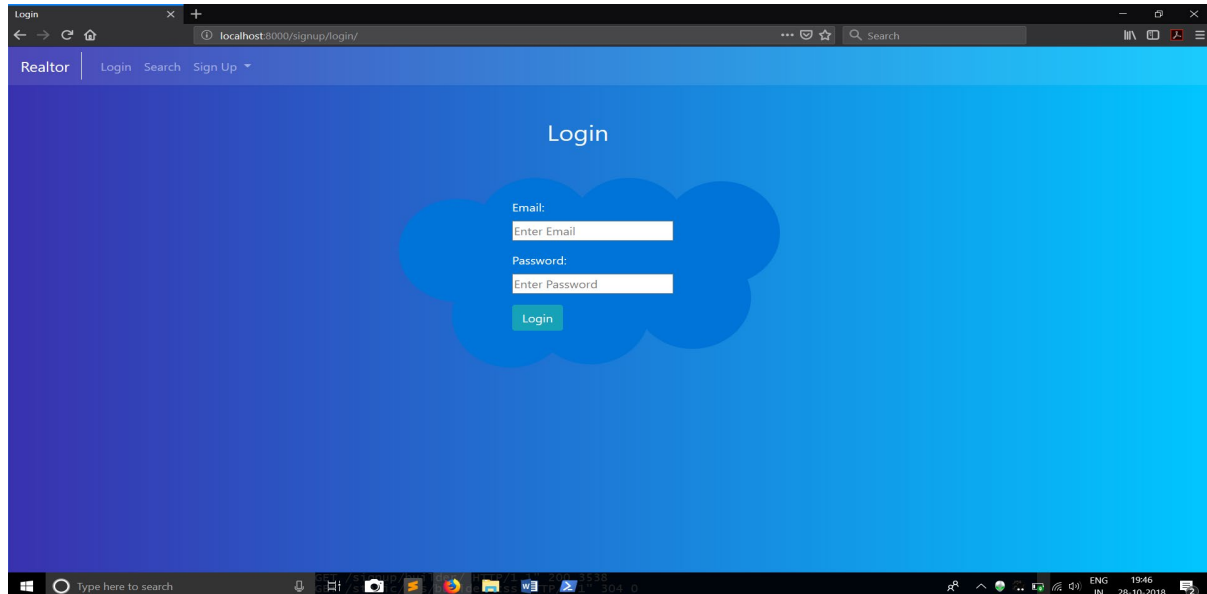
d) Signup Page

The sign up page is used for users to register to our database. There is a page for each actor in our system that accepts different fields of input



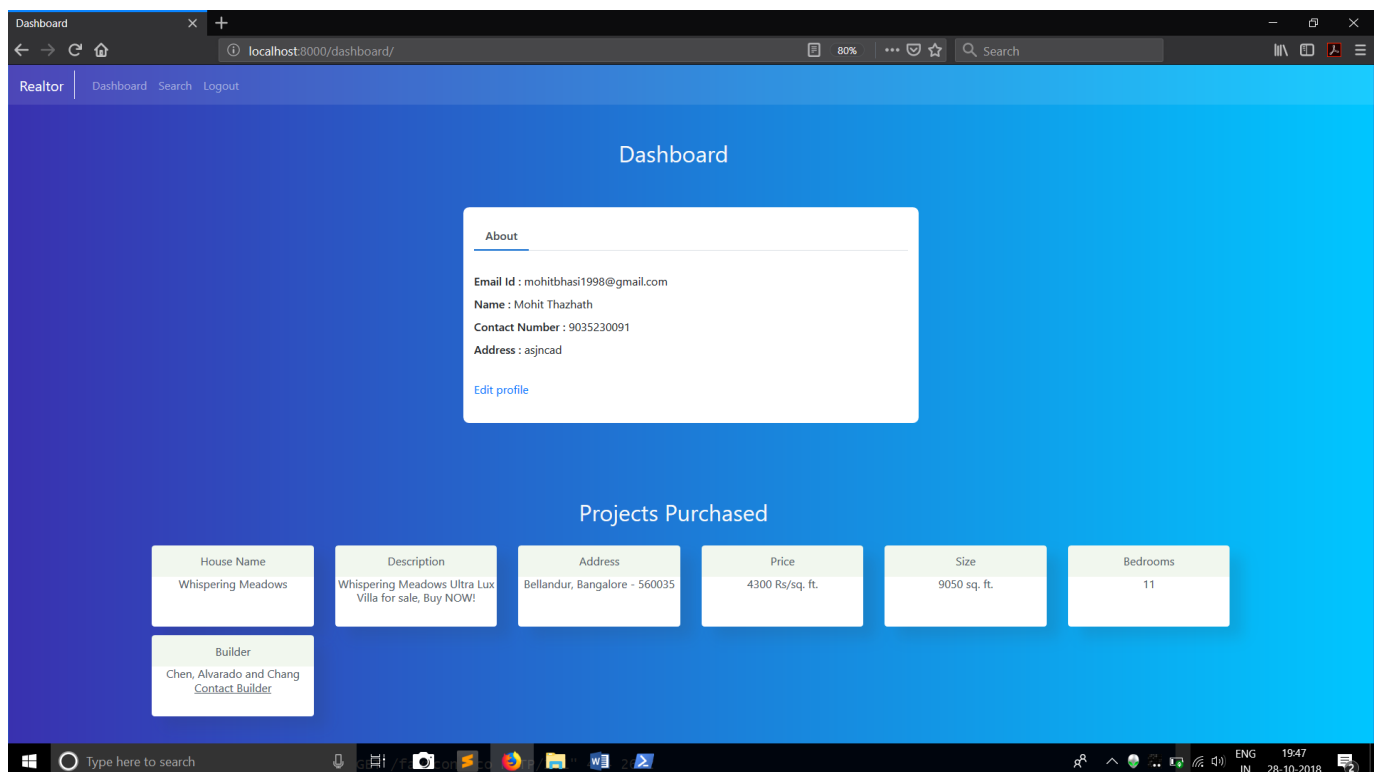
e) Login Page

The login page is the page where the user can login and check all the properties/projects owned/for sale by him/her.



f) Dashboard Page

Accessible once logged in, the dashboard is used by the users to manage their properties.



4.2 Security Measures:

1.SQL INJECTIONS

SQL injection is a type of attack where a malicious user is able to execute arbitrary SQL code on a database. This can result in records being deleted or data leakage.

Django's querysets are protected from SQL injection since their queries are constructed using query parameterization. A query's SQL code is defined separately from the query's parameters. Since parameters may be user-provided and therefore unsafe, they are escaped by the underlying database driver. Hence Raw SQL statements were not used through out the application.

3. BROKEN AUTHENTICATION & SESSION MANAGEMENT

Broken authentication and session management encompass several security issues, all of them having to do with maintaining the identity of a user. If authentication credentials and session identifiers are not protected at all times an attacker can hijack an active session and assume the identity of a user.

Django's stores it's session token in a hashed manner and is inaccessible to any other applications except the Django server which has a private key that can help decrypt the session token.

3. INSECURE DIRECT OBJECT REFERENCES

Insecure direct object reference is when a web application exposes a reference to an internal implementation object. Internal implementation objects include files, database records, directories, and database keys. When an application exposes a reference to one of these objects in a URL hackers can manipulate it to gain access to a user's personal data.

4.CROSS-SITE REQUEST FORGERY (CSRF)

CSRF attacks allow a malicious user to execute actions using the credentials of another user without that user's knowledge or consent.

Django has built-in protection against most types of CSRF attacks, providing you have enabled and used it where appropriate. However, as with any mitigation technique, there are limitations. For example, it is possible to disable the CSRF module globally or for particular views. You should only do this if you know what you are doing. There are other limitations if your site has subdomains that are outside of your control.

CSRF protection works by checking for a secret in each POST request. This ensures that a malicious user cannot simply "replay" a form POST to your website and have another logged in user unwittingly submit that form. The malicious user would have to know the secret, which is user specific (using a cookie)

CONCLUSION

Since this project has been designed exclusively as a project, certain complexities that do faced by any real life manual problem like total no. of database entries, address redundancy etc. are considered in this project. But enhancement to the project can easily be made without changing the current design and programming structure.

The end result of the project is a successful implementation of Realtor; a proposed system used to connect Home Buyers, Landlords and Builders. The whole application was built using Python and the Django framework. The data was stored using the open source MySQL platform and a lot of constraints were kept in mind while creating the structure of the tables.

Security measures were taking into consideration, many of the security measures provided by Django were implemented into the project without fail. One such mechanism is the CSRF Token, Django provides a built in token for each form submission which identifies the request as a unique one. This prevents users to get session hacked or hacked from other websites which may be requesting user information.