**Pick A Brick(Real Estate Website)**

A

Minor Project-I Report

*Submitted in partial fulfillment of the requirement for the award of*

*Degree of*

**BACHELOR OF TECHNOLOGY**

In

**Computer Science & Engineering**

Submitted to



**RAJIV GANDHI PRADYOGIKI VISHWAVIDHYALAYA,**

**BHOPAL (M.P.)**

Submitted by

**Srishti Gupta** 0225CS181055

**Vaishnavi Lakhera** 0225CS181059

**Prashansha Mishra** 0225CS181032

Under the Supervision of

**Prof. Rajendra Arakh**

Designation, Department

GNCSGI, Jabalpur



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**GLOBAL INSTITUTE OF ENGINEERING, JABALPUR**

**Session 2018-2022**

**GLOBAL INSTITUTE OF ENGINEERING, JABALPUR**

****

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**CERTIFICATE**

This is to certify that **Srishti Gupta, 0225CS181055, Vaishnavi Lakhera, 0225CS181059, Prashansha Mishra, 0225CS181032** of B.Tech Third year, **Computer Science & Engineering** have completed their minor project entitled **“Pick A Brick(Real Estate Website)”** during the year 2020 under my guidance and supervision.

I approve the project for the submission for the partial fulfillment of the requirement for the award of degree **Bachelor of Technology in** **Computer Science & Engineering.**

Guided & Approved by:

**Prof. Rajendra Arakh**

Professor, CSE

GNCSGI, Jabalpur

**(Prof. Sumit Nema) (Dr. Rajiv Khatri)**

HOD, CSE Department Director

GNCSGI, Jabalpur GNCSGI, Jabalpur

**GLOBAL INSTITUTE OF ENGINEERING, JABALPUR**

****

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**DECLARATION BY CANDIDATE**

We, **Srishti Gupta, Vaishnavi Lakhera, Prashansha Mishra,** students of Bachelor of Technology, Computer Science & Engineering Branch, Global Institute of Engineering, Jabalpur hereby declare that the work presented in this Minor project is outcome of our own work, is bonafide, correct to the best of my knowledge and this work has been carried out taking care of Engineering Ethics. The work presented does not infringe any patented work and has not been submitted to any University for the award of any degree or professional diploma.

**Srishti Gupta** 0225CS181055

**Vaishnavi Lakhera** 0225CS181059

**Prashansha Mishra** 0225CS181032

**ACKNOWLEDGEMENT**

At the outset, we would like to thank our guide and advisor, **Prof. Rajendra Arakh, Professor, Computer Science and Engineering** for giving us an opportunity to work on this challenging topic and providing us ample and valuable guidance throughout the Project.

We also greatly indebted to **Prof. Sumit Nema, Head Of Department, Computer Science and Engineering, GNCSGI,** without his encouragement and constant guidance we could not have finished this project. He was always a source of inspiration and motivator for innovative ideas during the entire span of this work.

We equally grateful to **Dr. Rajiv Khatri, Director, GNCSGI,** for providing all the necessary resources to carry out this Project work. We would like to thank all staff members of CSE, GNCSGI. And friends, for their support.

We will be failing in our duty if we don’t acknowledge the people behind this work to give us moral and psychological support. We hereby, would like to thank our parents for their endless care and constant support throughout our project work.

Srishti Gupta, 0225CS181055

Vaishnavi Lakhera, 0225CS181059

Prashansha Mishra, 0225CS181032

**GLOBAL INSTITUTE OF ENGINEERING, JABALPUR**

****

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**CERTIFICATE OF APPROVAL**

The Project work entitled **“Pick A Brick (A Real Estate Website)”** being submitted by **Srishti Gupta, 0225CS181055, Vaishnavi Lakhera, 0225CS181059, Prashansha Mishra, 0225CS181032,** have been examined by us andis hereby approved for the award of degree “**Bachelor of Technology (Computer Science & Engineering)”.**

(Internal Examiner) (External Examiner)

Date: Date:

**ABSTRACT**

Pick A Brick – This project is a property dealing website, which is implemented by using HTML, CSS framework- Bootstrap and JavaScript language for frontend, Business logic layer in Java Language and Advance Java and MySQL for backend.

It is basically an online platform which provides services of offering sale, purchase and lease or rent of plots, flats or bungalows in residential, commercial or industrial areas. The user can purchase, sale or lease the property in the form of plots, semi-built or built form. This project is used in property dealings that include commercial property dealings, residential property dealings, industrial property dealing and plots, flats, bungalows etc. The whole information is saved in the database which is created by using MySQL Workbench server 8.0 and the project is deployed in AWS Server.

Only the Property Dealer can have control over this website. This project is designed to save the data of all those persons who will sell, lease, rent or buy any kind of property like buildings, flats, plots etc. which will reduce manual work and helps the dealer to save the records efficiently. It provides the simplest, cheapest and an less time and energy consuming way of Property Dealing.

**CONTENTS**

1. Introduction
   1. Project Overview
   2. Project Deliverables
2. Project Organization
   1. Software Process Model
   2. Roles and Responsibilities
   3. Tools And Techniques
3. Project Management Plan
   1. Task-n and Description
   2. Deliverables and Milestones
   3. Resources Needed
   4. Dependencies and Constraints
4. Software Design Description
   1. Design Overview/ Algorithms
   2. System Design
      1. Use Case Diagram
      2. Data Flow Diagram
      3. E-R Diagram
   3. User Interface Design
      1. Screen Images
5. Conclusion
6. References

**INTRODUCTION**

* **Project Overview:**

Pick a brick – property dealing website, this project is a user friendly website developed in programming language HTML, CSS, JavaScript as front end, Core Java and Advance Java as business logic and SQL server database as back end. The project is built by following the MVC Architecture.

The Real Estate website is built using a layered architecture where the total functionality can be divided into layers having different functionalities. The main layers include the database access layer, business logic layer and the presentation layer. Thus, this application follows the 3-Tier Architecture.

This project is designed to save the data of all those persons who will sell, lease, rent or buy any kind of property like buildings, flats, plots etc. which will reduce manual work and helps the dealer to save the records efficiently. It provides the simplest, cheapest and an less time and energy consuming way of Property Dealing.

The scope of this project “Real Estate Website” is to enable the buyers to search for property listings online. The motive of developing this site is to design a feature rich platform which can make the search of property dealings, residential property dealings, industrial property dealing and plots, flats, bungalows etc. an easy task.

* **Project Deliverables:**
* In old days, the active Property Dealers may have to keep records of the properties manually and also for common man, it was hard to find the property of their own choice and in their area. This method is quiet time consuming and less efficient. Also there are more chances of mistakes by keeping the records manually as human beings are habitual of doing mistakes. So with the help of this, Property Dealing Website, the chances of mistakes becomes very few.
* This website allows the seller to sell or to let their owned property, also to the buyers to buy, rent or lease a property.
* The user has to be a verified user and for verification they have to verify their email address and mobile number.
* Buyers have the facilities of searching for a property anywhere in the world and buyer can then add the any property to its wish list, can bid on any property with the amount specified and can also directly buy the property.
* The seller have the facility to add a property to sell or to rent a property and providing every specific details of the property such as address, features, construction status, pictures and much more.

Seller can see who has put his property in wish list and who has put a bid on his property.

* Another amazing feature and service that this site provides is that it enables chat online option to all the bidders (buyers who has put a bid) and sellers for a particular property.
* The website provide very efficient method of keeping the records of properties sold as well as unsold properties. It consumes very less time as compared to manual method.
* By allowing the site to access your location, it will start giving you all the near by property suggestions, thus making it easier for the user who is willing to own a property in their own location and giving user personalized services

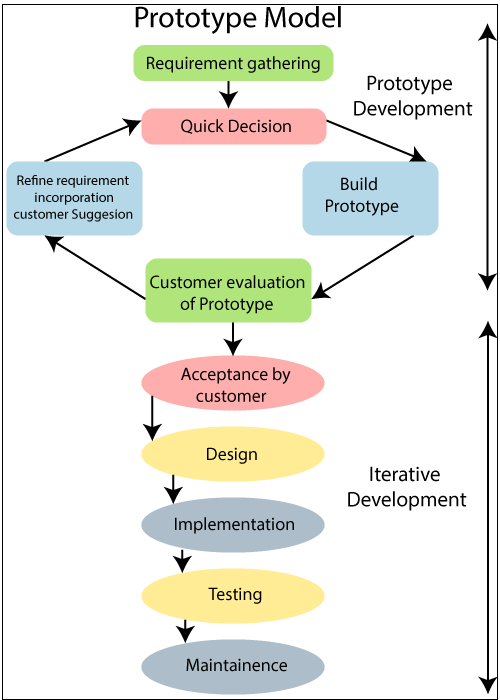
**Project Organizations**

* **Software Process Model:**
* **Prototype Model –**

Prototyping process is an important part of software development. One of the most important parts of software development is project design.

Software project designing as a process of project creation can be divided in two large parts (very conditional): design of the functionality and design of user interface. Prototypes can be different: paper, presentation, imitation, etc., up to exact correspondence to the future program.

The prototyping model allows you to create a prototype of a software product before or during the stage of compiling requirements for the software product. Potential users work with this prototype, identifying its strengths and weaknesses, the results are reported to the developers of the software product. Thus, feedback is provided between users and developers, which is used to change or correct the specification of requirements for the software product. As a result of this work, the product will reflect the real needs of users.



* Requirement gathering and analysis:

A prototyping model starts with requirement analysis. In this phase, the requirements of the system are defined in detail. During the process, the users of the system are interviewed to know what is their expectation from the system.

* Quick design:

The second phase is a preliminary design or a quick design. In this stage, a simple design of the system is created. However, it is not a complete design. It gives a brief idea of the system to the user. The quick design helps in developing the prototype.

* Build prototype:

In this phase, an actual prototype is designed based on the information gathered from quick design. It is a small working model of the required system.

* Initial user evaluation:

In this stage, the proposed system is presented to the client for an initial evaluation. It helps to find out the strength and weakness of the working model. Comment and suggestion are collected from the customer and provided to the developer.

* Refining prototype:

If the user is not happy with the current prototype, you need to refine the prototype according to the user's feedback and suggestions. This phase will not over until all the requirements specified by the user are met. Once the user is satisfied with the developed prototype, a final system is developed based on the approved final prototype.

* Implement product and maintain:

Once the final system is developed based on the final prototype, it is thoroughly tested and deployed to production. The system undergoes routine maintenance for minimizing downtime and prevent large-scale failures.

Thus, we choose to use this model as web interfaces have a very high amount of interaction with end users, are best suited for Prototype model. It might take a while for a system to be built that allows ease of use and needs minimal training for the end user.

* **Roles and Responsibility:**
* Team member 1 - Srishti Gupta:

Design User Interface prototype of project i.e. our website Pick A Brick, Also done a major part of project i.e. develop business logic layer of project, worked on Aws services for deployment like connecting the deployed site with the domain name, using RDS services, getting SSL certificate for the website etc. Unit testing has also been performed.

* Team Member 2 - Vaishnavi Lakhera:

Designed DBMS schema for Pick A Brick, Run and tested all the DBMS Query in backend which will going to fetch the data and store it safely, created content for the website, set up DNS records, created and register Domain for website. Integration testing has also been done , worked on API services as we used Tomtom API for map locations .

* Team Member 3 - Prashansha Mishra:

Worked on 3rd party email services for the verification of users.

Also worked on OTP services, helped in designing part of website like themes, etc to give the attractive look. Also collected the data and stated the requirements needs to be accomplished.

* **Tools and Techniques:**

The latest tools and technologies involved in building this website are:

Java Servlet API, Java Server Pages Technology, **JavaServer Pages Standard Tag Library,** Java Message Service API, JDBC API, HTML CSS framework - Bootstrap, SQL server-8.0, Tomcat 9.0, AJAX and Java Script, Microsoft Visual Studio, Amazon Web Services.

* Java Servlet API:

The Java Servlet API lets you define HTTP-specific classes. A servlet class extends the capabilities of servers that host applications that are accessed by way of a request-response programming model. Although servlets can respond to any type of request, they are commonly used to extend the applications hosted by web servers. For instance, you might use a servlet to get the text input from an online form and print it back to the screen in an HTML page and format, or you might use a different servlet to write the data to a file or database instead. A servlet runs on the server side -- without an application GUI or HTML user interface (UI) of its own. Java Servlet extensions make many web applications possible.

The [javax.servlet](https://www.oracle.com/technetwork/java/overview-137084.html)and [javax.servlet.http](https://www.oracle.com/technetwork/java/overview-137084.html)packages provide the classes and interfaces to define servlets. HTML servlet classes extend the javax.servlet.http.HttpServlet abstract class, which provides a framework for handling HTTP protocol.

* Java Server Pages:

JavaServer Pages (JSP) technology provides a simplified, fast way to create dynamic web content. JSP technology enables rapid development of web-based applications that are server- and platform-independent. JSP technology lets you add snippets of servlet code directly into a text-based document. Typically, a JSP page is a text-based document that contains two types of text:

Static data, which can be expressed in any text-based format, such as HTML, Wireless Markup Language (WML), or XML

JSP technology elements, which determine how the page constructs dynamic content

The packages involved in creating JSP pages are javax.el, javax.servlet.jsp, javax.servlet.jsp.el, and javax.servlet.jsp.tagext, though you will rarely have to import these directly.

A JSP page can be as simple as a bit of HTML with one snippet of JSP code and the .jsp extension of the page name.

For instance, you can create a web site of JSP technology pages that use one snippet of code to include the header.html file, which contains the site navigation. This way, when you change a link to a button in the navigation, you make the change in only one file, and that file loads into all the pages on the site that have this code snippet:

**<%@ include file="header.html" %>**

That line of code works very much like a server-side include, if you are familiar with those. Because this web page is now a JSP page, you could also go on to add more Java technology code to create dynamic web content, such as polls, forms, ways to enter or retrieve data from a database, and so forth.

* Java Message Service API:

Web applications can use the JavaMail API to send email notifications. The API has two parts: an application-level interface that the application components use to send email and a service provider interface. Service providers implement particular email protocols, such as SMTP. Several service providers are included with the JavaMail API package, and others are available separately. The Java EE platform includes the JavaMail extension with a service provider that allows application components to send email. By combining Java technology with enterprise messaging, the Java Message Service (JMS) API provides a powerful tool for solving enterprise computing problems.

Enterprise messaging provides a reliable, flexible service for the exchange of business data throughout an enterprise. The JMS API adds to this a common API and provider framework that enables the development of portable message-based applications in the Java programming language. The parts components in turn can send messages to their own inventory and order components to update their inventories and to order new parts from suppliers and so forth.

The JMS API improves programmer productivity by defining a common set of messaging concepts and programming strategies that all JMS technology-compliant messaging systems will support.

* **JavaServer Pages Standard Tag Library:**

The JavaServer Pages Standard Tag Library (JSTL) encapsulates core functionality common to many JSP technology-based applications.

Instead of mixing tags from numerous vendors in your applications, you employ a single standard set of tags.

This standardization allows you to deploy your applications on any JSP container that supports JSTL and makes it more likely that the implementation of the tags is optimized.

JSTL has iterator and conditional tags for handling flow control, tags for manipulating XML documents, internationalization tags, tags for accessing databases using SQL, and tags for commonly used functions.

The packages you can access for using JSTL are [javax.servlet.jsp.jstl.core](https://docs.oracle.com/javaee/5/api/javax/servlet/jsp/jstl/core/package-summary.html),[javax.servlet.jsp.jstl.fmt](https://glassfish.dev.java.net/nonav/javaee5/api/javax/servlet/jsp/jstl/fmt/package-summary.html), [javax.servlet.jsp.jstl.sql](https://glassfish.dev.java.net/nonav/javaee5/api/javax/servlet/jsp/jstl/sql/package-tree.html) and [javax.servlet.jsp.jstl.tlv](https://docs.oracle.com/cd/E17802_01/webservices/webservices/docs/1.5/api/javax/servlet/jsp/jstl/tlv/package-summary.html).

* JDBC API:

The JDBC API allows you invoke database SQL commands from Java programming language methods. You can use the JDBC API in a servlet, JSP technology page, or an enterprise bean when you need to access the database.

The JDBC API has two parts: an application-level interface that application components use to access a database and a service provider interface to attach a JDBC driver to the Java EE platform.

* Microsoft Visual Studio:

Microsoft Visual Studio is an [integrated development environment](https://en.wikipedia.org/wiki/Integrated_development_environment) (IDE) from [Microsoft](https://en.wikipedia.org/wiki/Microsoft). It is used to develop [computer programs](https://en.wikipedia.org/wiki/Computer_program), as well as [websites](https://en.wikipedia.org/wiki/Web_site), [web apps](https://en.wikipedia.org/wiki/Web_app), [web services](https://en.wikipedia.org/wiki/Web_service) and [mobile apps](https://en.wikipedia.org/wiki/Mobile_app). Visual Studio supports 36 different [programming languages](https://en.wikipedia.org/wiki/Programming_language) and allows the code editor and debugger to support (to varying degrees) nearly any programming language, provided a language-specific service exists. Built-in languages include [C](https://en.wikipedia.org/wiki/C_(programming_language)), C++ [,](https://en.wikipedia.org/wiki/C%2B%2B) [JavaScript](https://en.wikipedia.org/wiki/JavaScript), [TypeScript](https://en.wikipedia.org/wiki/TypeScript), [HTML](https://en.wikipedia.org/wiki/HTML), and [CSS](https://en.wikipedia.org/wiki/Cascading_Style_Sheets). Support for other languages such as [Python](https://en.wikipedia.org/wiki/Python_(programming_language)), [Ruby](https://en.wikipedia.org/wiki/Ruby_(programming_language)), [Java](https://en.wikipedia.org/wiki/Java_(programming_language)) and may more.

* **Front End:**

HTML and CSS: [HTML](https://www.w3.org/html/) (the Hypertext Markup Language) and [CSS](https://www.w3.org/Style/CSS/) (Cascading Style Sheets) are two of the core technologies for building Web pages. HTML provides the structure of the page, CSS the (visual and aural) layout, for a variety of devices.

Along with [graphics](https://www.w3.org/standards/webdesign/graphics) and [scripting](https://www.w3.org/standards/webdesign/script), HTML and CSS are the basis of building Web pages and Web Applications. [HTML](https://www.w3.org/html/) is the language for describing the structure of Web pages.

HTML gives authors the means to:

Publish online documents with headings, text, tables, lists, photos, etc.

Retrieve online information via hypertext links, at the click of a button.

Design forms for conducting transactions with remote services, for use in searching for information, making reservations, ordering products, etc. Include spread-sheets, video clips, sound clips, and other applications directly in their documents.

Bootstrap 4: Bootstrap is a [free and open-source](https://en.wikipedia.org/wiki/Free_and_open-source) [CSS framework](https://en.wikipedia.org/wiki/CSS_framework) directed at responsive, [mobile-first](https://en.wikipedia.org/wiki/Responsive_web_design#Mobile_first,_unobtrusive_JavaScript,_and_progressive_enhancement) [front-end web development](https://en.wikipedia.org/wiki/Front-end_web_development). It contains [CSS](https://en.wikipedia.org/wiki/CSS)- and (optionally) [JavaScript](https://en.wikipedia.org/wiki/JavaScript)-based design templates for [typography](https://en.wikipedia.org/wiki/Web_design#Typography), [forms](https://en.wikipedia.org/wiki/Form_(HTML)), [buttons](https://en.wikipedia.org/wiki/Button_(computing)#HTML), [navigation](https://en.wikipedia.org/wiki/Web_navigation#Local_website_navigation), and other interface components.

JavaScript: **JavaScript (JS)** is a lightweight, interpreted, or [just-in-time](https://en.wikipedia.org/wiki/Just-in-time_compilation) compiled programming language with [first-class functions](https://developer.mozilla.org/en-US/docs/Glossary/First-class_Function). While it is most well-known as the scripting language for Web pages, [many non-browser environments](https://en.wikipedia.org/wiki/JavaScript#Uses_outside_Web_pages) also use it, such as [Node.js](https://developer.mozilla.org/en-US/docs/Glossary/Node.js).

Ajax: AJAX stands for **A**synchronous **J**avaScript **A**nd **X**ML. In a nutshell, it is the use of the [XMLHttpRequest](https://developer.mozilla.org/en/DOM/XMLHttpRequest) object to communicate with servers. Ajax is the primary technique that websites use to retrieve data today, but with XML largely substituted for JSON. It can send and receive information in various formats, including JSON, XML, HTML, and text files. AJAX’s most appealing characteristic is its "asynchronous" nature, which means it can communicate with the server, exchange data, and update the page without having to refresh the page.

Async/await is a new syntax (borrowed from .NET and C#) that allows us to compose Promises as though they were just normal synchronous functions without callbacks. It's a fantastic addition to the JavaScript language, added last year in JavaScript ES7, and can be used to simplify pretty much any existing JS application.

* **Back End:**

SQL server-8.0: MySQL is open-source, cross-platform relational database management server developed by Swedish company “MySQL AB” and later acquired by Oracle corporation. MySQL is offered as an open-source MySQL community server edition and enterprise server edition.

Amazon Web Services: Amazon Web Services (AWS) is a subsidiary of [Amazon](https://en.wikipedia.org/wiki/Amazon.com) providing [on-demand](https://en.wikipedia.org/wiki/Software_as_a_service) [cloud computing](https://en.wikipedia.org/wiki/Cloud_computing) [platforms](https://en.wikipedia.org/wiki/Computing_platform) and [APIs](https://en.wikipedia.org/wiki/Application_programming_interface) to individuals, companies, and governments, on a metered pay-as-you-go basis.

AWS Elastic Beanstalk is a service that lets you define an environment for common application types, and deploy code into it. The Beanstalk template is dependent on the VPC, and optionally can be used with the bastion, RDS, or Aurora template.

[Amazon Relational Database Service (RDS)](https://aws.amazon.com/rds/) is a service for running relational databases without having to manage the server software, backups, or other maintenance tasks.

AWS Certificate Manager is a service that lets you easily provision, manage, and deploy public and private Secure Sockets Layer/Transport Layer Security.

Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides secure, resizable compute capacity in the cloud.

Project Management Plan

* Task-n and Description:
* **Designing a Database and creating a table:**

Gather the requirements and define the objective of your database, gather the data that are needed to be stored in the database. Divide the data into subject-based tables. Also define the relationship between the tables.

* **Designing a prototype of the User Interface:**

**A website prototype** takes it to a whole new level by adding interactions and animation and giving a look and feel of what an end product will look like. The key reason for creating the prototype is to get feedback to make sure it is going in the right direction. It allows to fix any areas which receive negative feedback, and it allows for discussion about what is really needed in the final build and what could be removed.

* **Connecting a front end with the backend using business logic:**

Following the MVC architecture the different Java classes and .jsp and servlet pages are divided in 3 parts i.e models containing plain java classes having code and queries that will interact with database, controllers having the servlet classes that will act as the business logic. The **Business Layer** contains the application and business logic that facilitate the functions and services that the application provides.

And view having the .jsp pages which are the view of users ie they are the front end part.

* **Making asynchronous request call possible:**

In asynchronous programs, you can have two lines of code (L1 followed by L2), where L1 schedules some task to be run in the future, but L2 runs before that task completes.

Using promises and async/await to handle asynchronous call and also setTimeout() function to make the fetching of data possible dynamically and thus making the requesting of data independent.

* **Using Map API to track users location and make nearby suggestion:**

For tracking the users location and the showing the nearby suggestion to the user just to make our site more interactive and user friendly and easy to use and navigate, we used the Tomtom Map API services and tracked the users location and also the property’s location via their address thus making it more dynamic and providing users what they need in their feed.

* **Deploying the website to the AWS server:**

AWS Amplify provides fully managed hosting for static websites and web apps.

Amplify’s hosting solution leverages Amazon CloudFront and Amazon S3 to deliver your site assets via the AWS content delivery network.

We have deployed our site using Elastic BeanStalk and created the online database using RDS services and finally using Route-53 services the site is deployed on AWS server.

* **Getting Domain name:**

To make our site uniquely identifiable and have our site get called by its own name, we get our own domain name registered and attached our site to the domain name. For getting the domain name we have used freenom.com site services.

* **Getting SSL certificate for the website:**

SSL certificate helps protect your website data, it’s actually a requirement for accepting payments online. Using the normal HTTP protocol means this information can be hijacked by hackers. This is where SSL or HTTPS comes in. Using Certificate Manager of the AWS we successfully got the SSL certificate over HTTPs for our site.

* **Enabling third party Email Services:**

To be able to send the verification mail to our users during their registration, we need email services. Using a third-party host, you don't need to worry about the configuration and troubleshooting of the mailing service. No need to setup your SSL certificate for your server.

Thus, we have used Simple Mail Services to send the mail and makin our verification process successful.

* Deliverables and Milestone:
* Planned approach towards working: - The working in the organization will be well planned and organized. The data will be stored properly in data stores, which will help in retrieval of information as well as its storage.
* Accuracy: - The level of accuracy in the proposed system cannot be decided. Because here user buy and another user build the home. There is no guarantee.
* Reliability: - The reliability of the proposed system will be high due to the above stated reasons. The reason for the increased reliability of the system is that now there will be proper storage of information.
* The system is very useful for the companies or builders that can post and edit their properties and their personal info and admin can monitor records of all of them.
* The system is also useful which also keeps track of Account details of buyers and Investors.
* This website allows the seller to sell or to let their owned property, also to the buyers to buy, rent or lease a property.
* The user has to be a verified user and for verification they have to verify their email address and mobile number.
* Here user buy different property & sell them to the system. Buyer user property & builder verify each-other & make reliable communication to each other.
* User can visit the home page of real estate in which first the introduction of our site mention first. The registered user can login from the login module.
* Here guest can register free account to sell and buy property &buyer verify each- other & make reliable communication to each other. User can search the property and it can select the type of property and its budget and also find the location of property.
* Buyers have the facilities of searching for a property anywhere in the world and buyer can then add the any property to its wish list, can bid on any property with the amount specified and can also directly buy the property.
* The seller have the facility to add a property to sell or to rent a property and providing every specific details of the property such as address, features, construction status, pictures and much more.
* Seller can see who has put his property in wish list and who has put a bid on his property.
* Improve requirement accuracy: Industry-unique collaborative storyboarding improves accuracy by promoting effective communication.
* Another amazing feature and service that this site provides is that it enables chat online option to all the bidders (buyers who has put a bid) and sellers for a particular property.
* The website provide very efficient method of keeping the records of properties sold as well as unsold properties. It consumes very less time as compared to manual method.
* By allowing the site to access your location, it will start giving you all the nearby property suggestions, thus making it easier for the user who is willing to own a property in their own location and giving user personalized services.
* Our system must save time and money: Accurate upfront software requirements definition helps ensure your team works on the business problems that matter most.
* Reduce rework: Early validation and agreement by users means development and quality teams spend less time on rework.
* Resources Needed:
* HARDWARE REQUIREMENTS:

1. PROCESSOR: Intel i-5 Processor, AMD Ryzen 5 etc.

2. RAM: 4GB or higher

3. HARD DISK SPACE: 500GB or higher

4. SYSTEM: 32-bit Operating System or 64-bit Operating System

* SOFTWARE REQUIREMENTS:

1. WEB BROWSER: Google Chrome, Microsoft Edge, Mozilla Firefox

2. SERVER: Tomcat 9.0 (Local) or AWS Server

3. ENVIRONMENT: Microsoft Visual Studio.Net

4. DATABASE: SQL SERVER 8.0

5. PLATFORM: Windows

* Dependencies and Constraints:
* Internet Connection should be there with minimum speed of 200kbps
* If running of local server then Tomcat 9.0 version should be installed.
* Database should be MySQL Server 8.0

Software Design Description

* Design Overview/Algorithms:
  + MVC Architecture:

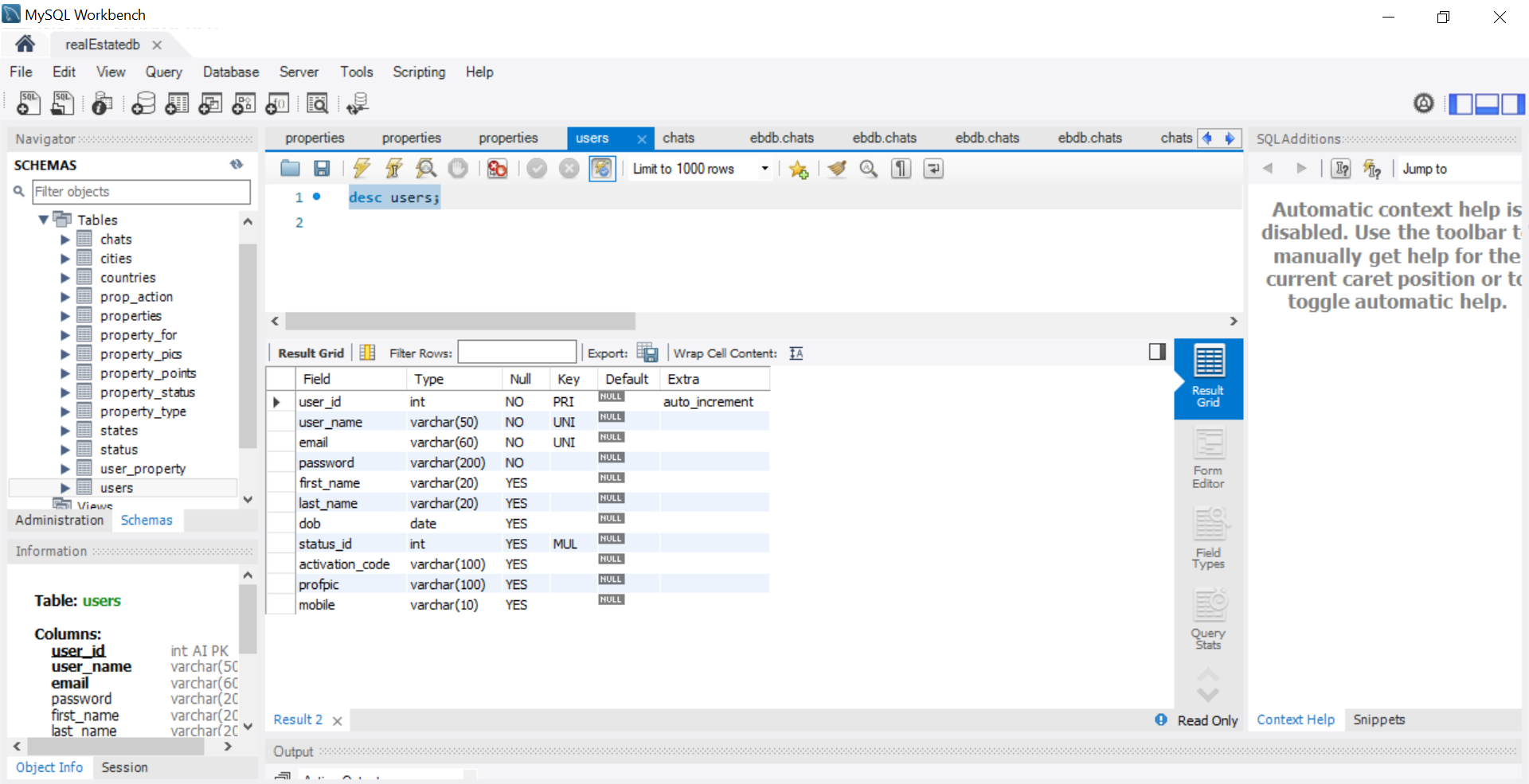
The Model-View-Controller (MVC) is an architectural pattern that separates an application into three main logical components: the model, the view, and the controller. Each of these components are built to handle specific development aspects of an application. MVC is one of the most frequently used industry-standard web development framework to create scalable and extensible projects.

* + **User Module**

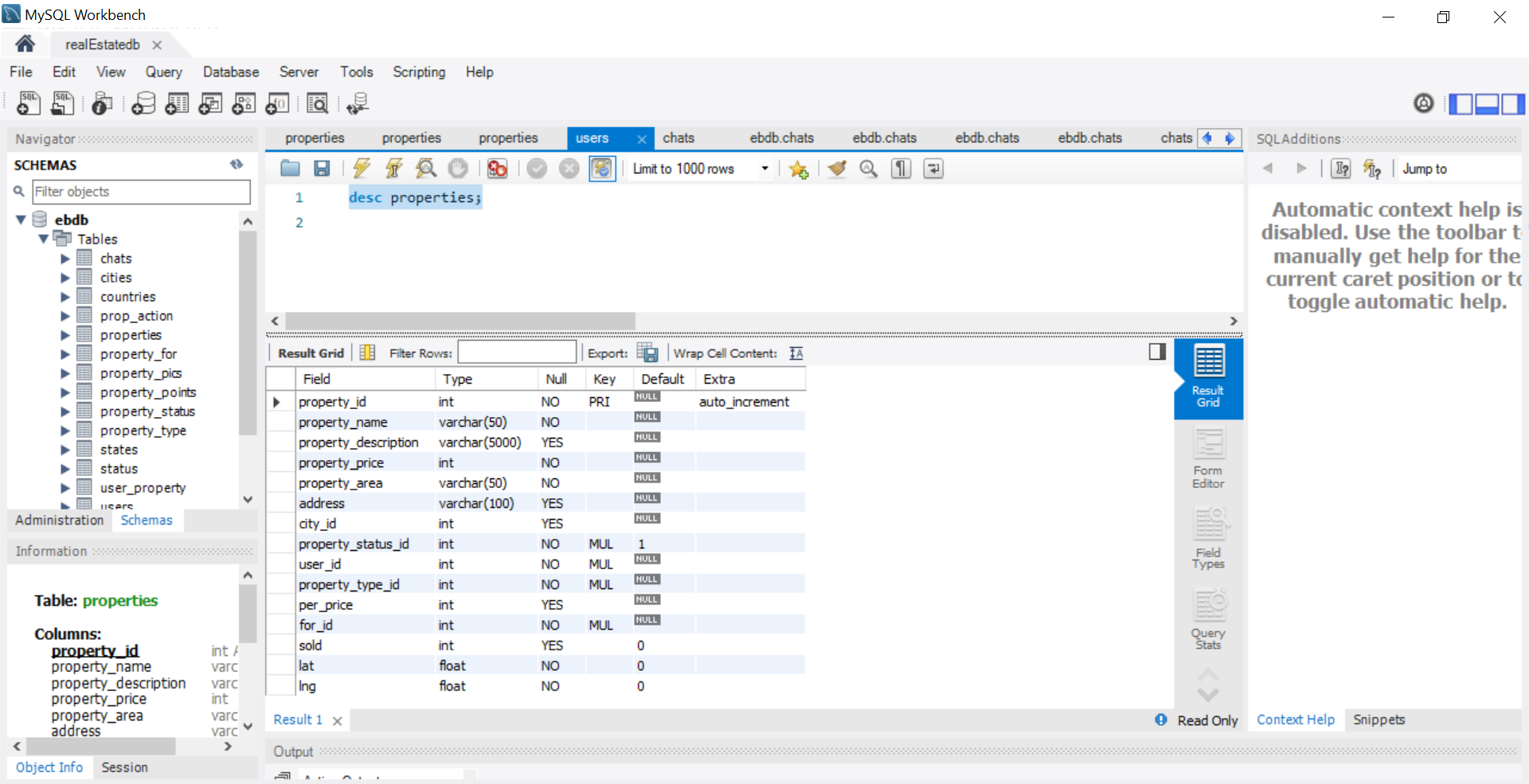
1. **Seller**
   * Home Page: Owner can view the home page of the real estate management system
   * About: Owner can view about us page.
   * Properties: Owner can view own and other owner properties they can view property type wise, Status wise and city wise.
   * Contact us: Owner can view contact us page.
   * Search: User can search for the properties adding filters and constraints.
   * My Account: In this, there is three sections:
   1. User Profile: In this section, the owner do the following activity
   2. Edit Profile: Owner can edit his/her own profile.
   3. Change Password: Owner can change his/her own password.
   4. Add Property: Owner can add his/her own property
   5. Seller Dashboard: Owner can see his/her own listed properties.
   6. Bids on your property: Owner can view receive enquiries against his/her own listed properties and also answer the enquiries.
   7. Wishlist: Owner can view the number of people added his property on their wish list.
   8. Chat Now: The seller can chat to all the bidders.
   9. Logout: Owner can logout from own account.
   10. Change Password: Owner can change his/her own password.
   11. Logout: Owner can logout from own account.
2. **Buyer**

* Home Page: User can view the home page of the real estate management system
* About: User can view about us page.
* Search Properties: User can view properties they can view property type wise, Status wise, price wise and city wise and put his/her enquiries against any property.
* Contact us: User can view contact us page.
* Your Bids: User can bid on seller’s property with the bidding amount and can see their bids.
* Your Wishlist: User can add seller’s property in their wish list and can see it on their wish list sections.
* Buy Property: User can buy the property and do the payment formalities.
* Chat Now: The user can chat to the seller on whose property he bids.
* My Account: In this, there is three sections:
  1. User Profile: In this section, Agents do the following activity
  2. Edit Profile: Agents can edit his/her own profile.
  3. Change Password: Agents can change his/her own password.
  4. Logout: Agents can logout from own account.
* Show Nearby: The user can see the nearby property by allowing to track it location.
* Database Design:

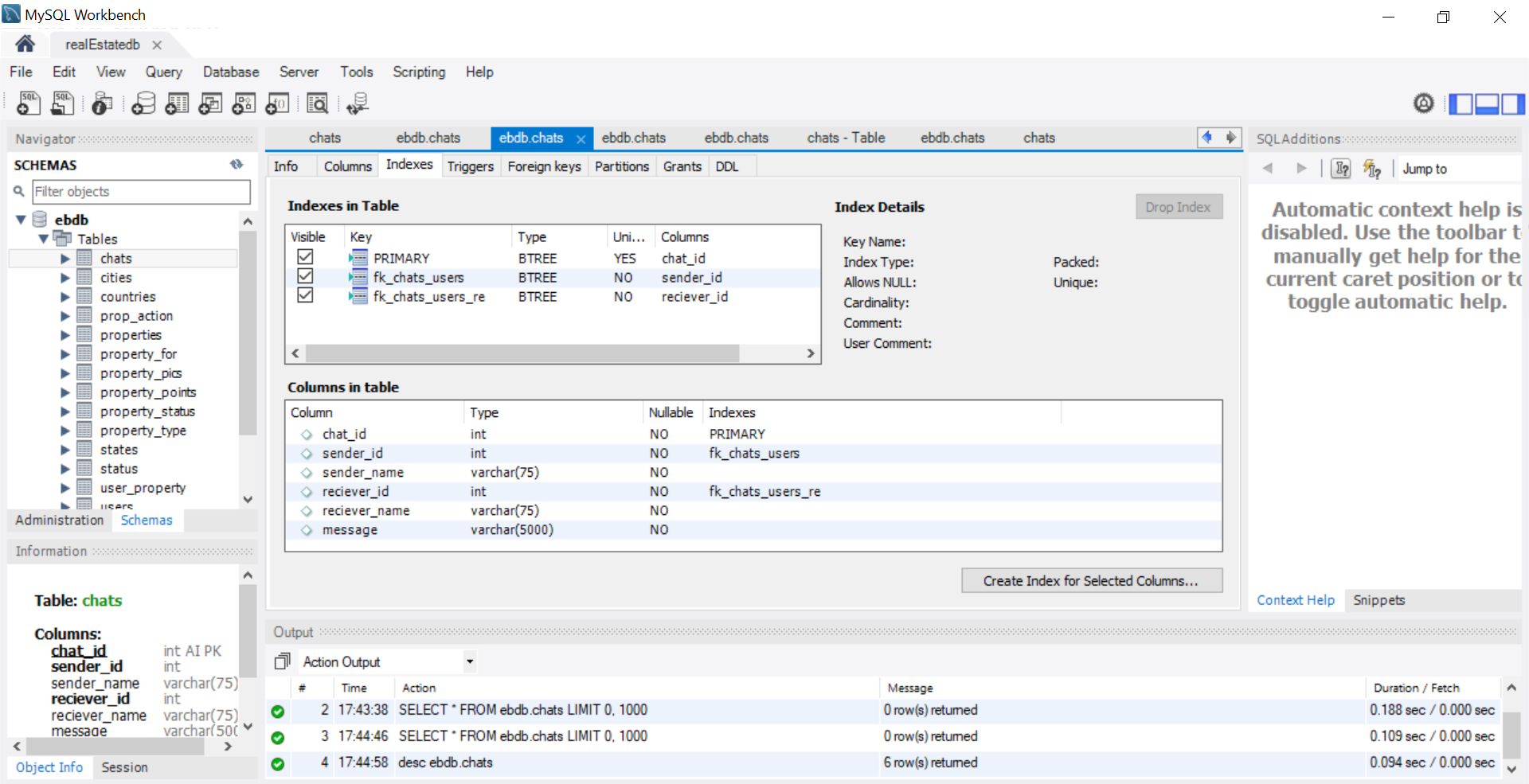
Users Table -



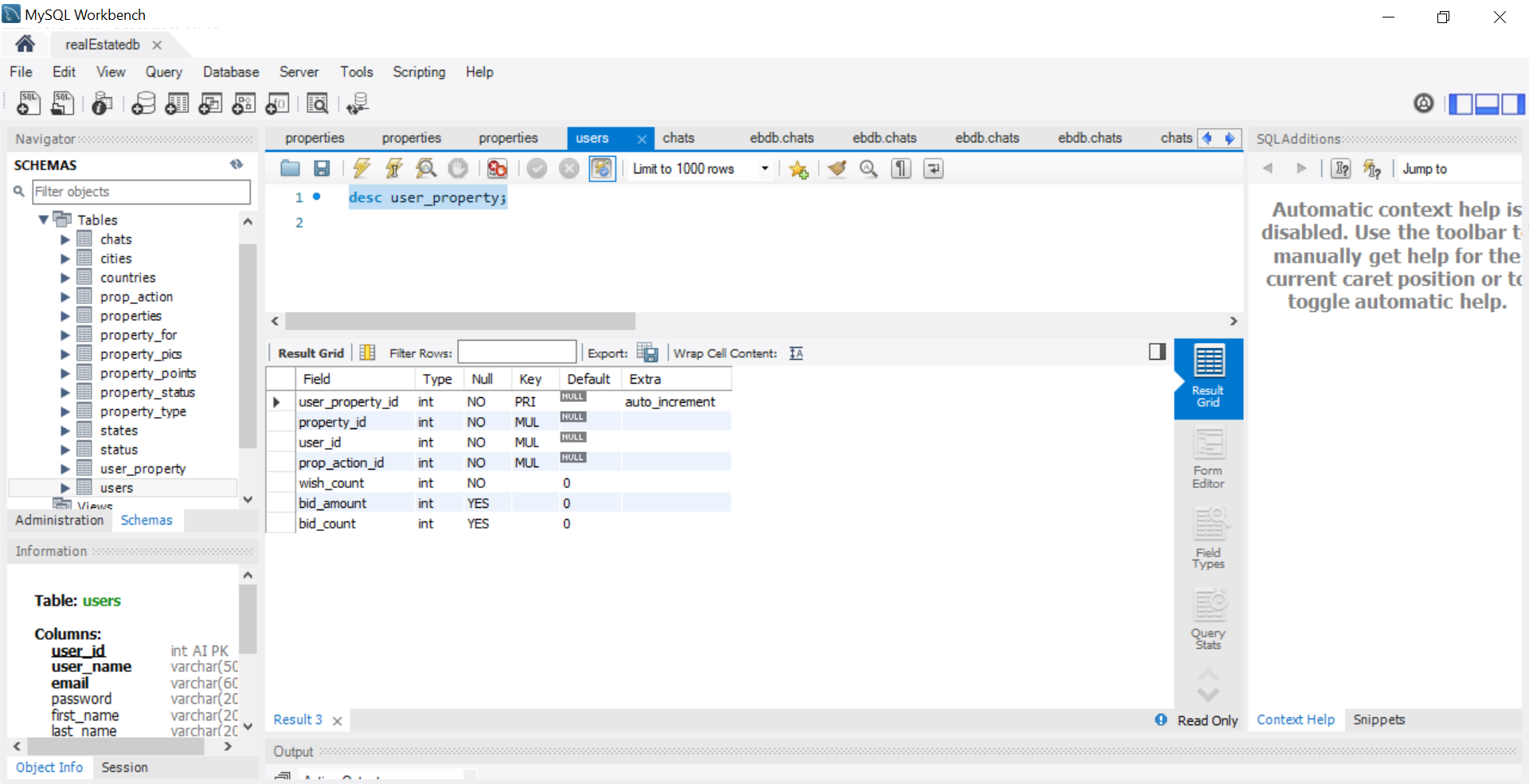
Properties Table

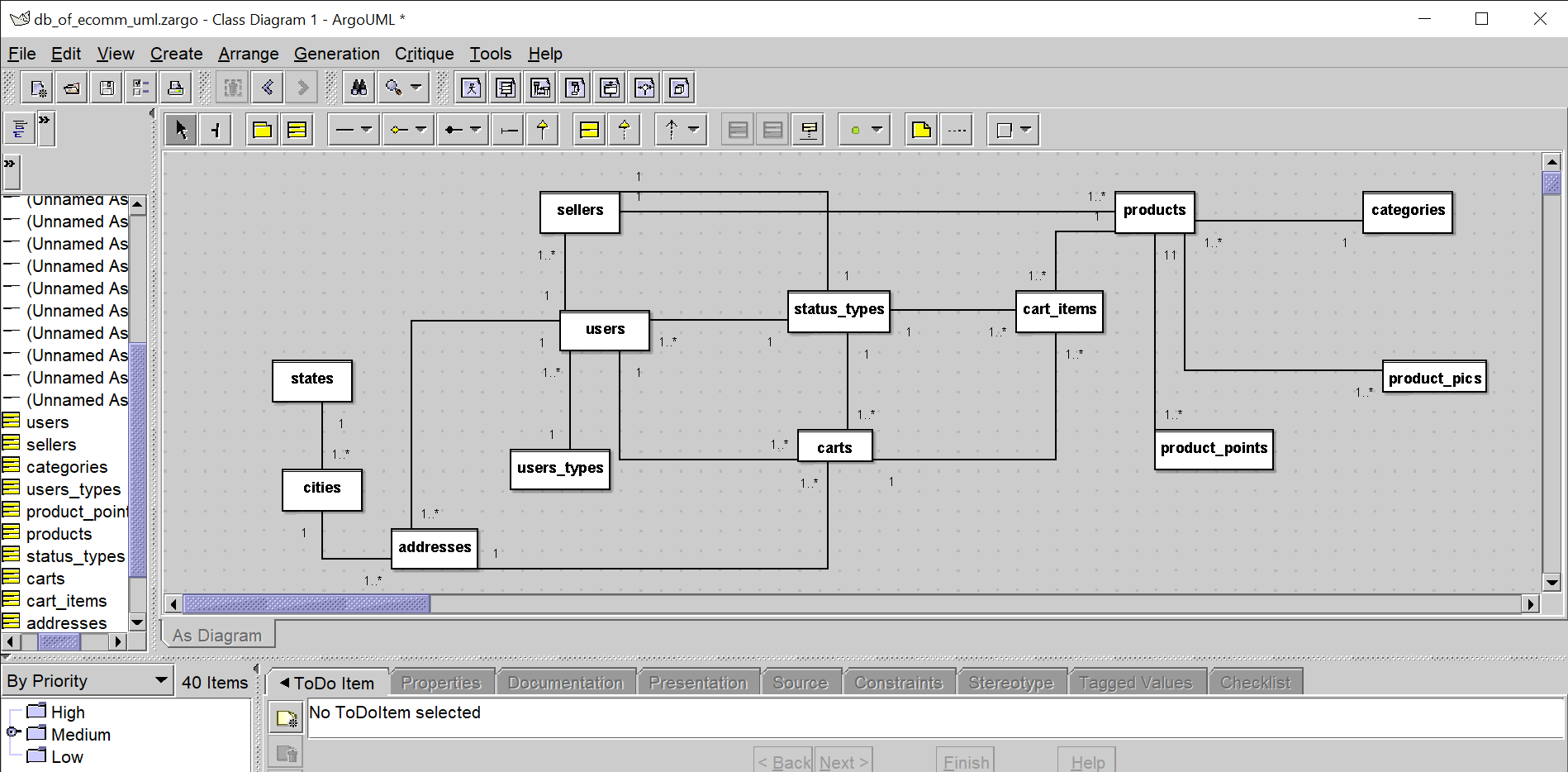


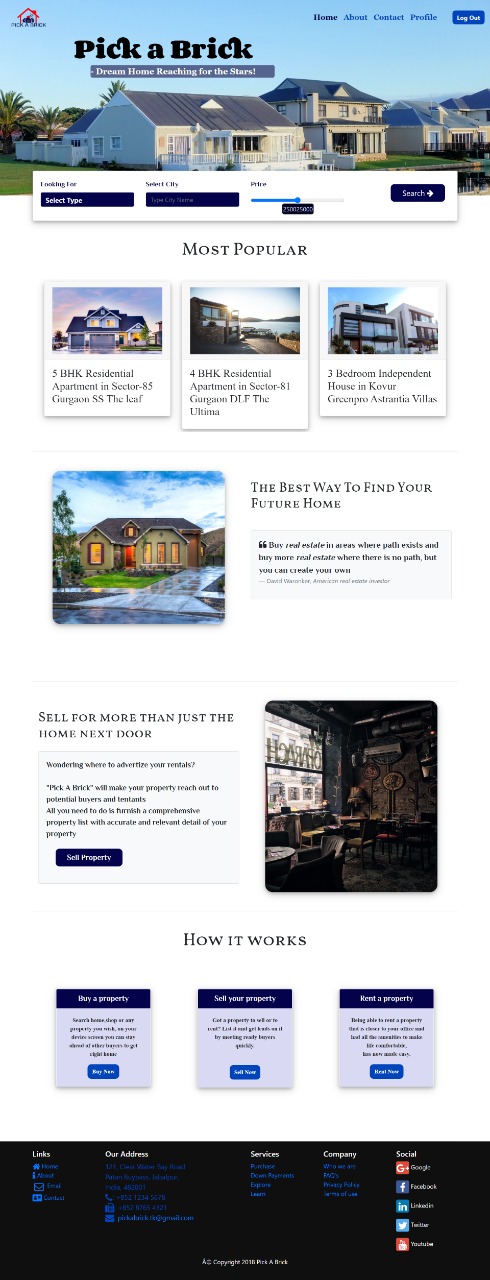
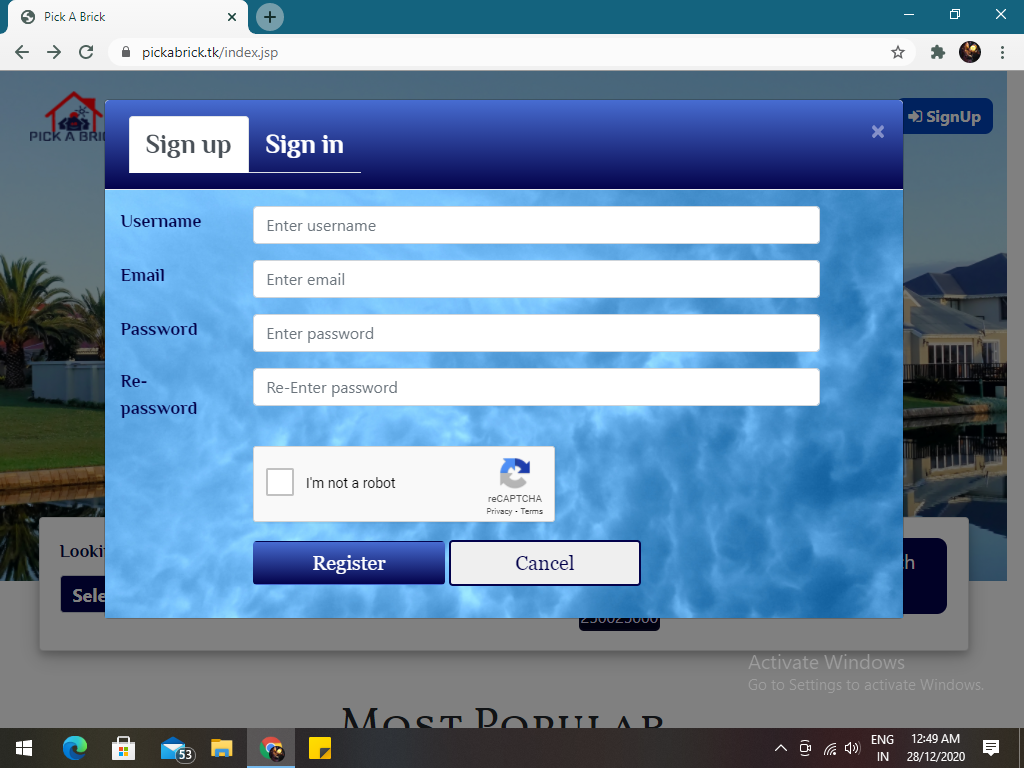
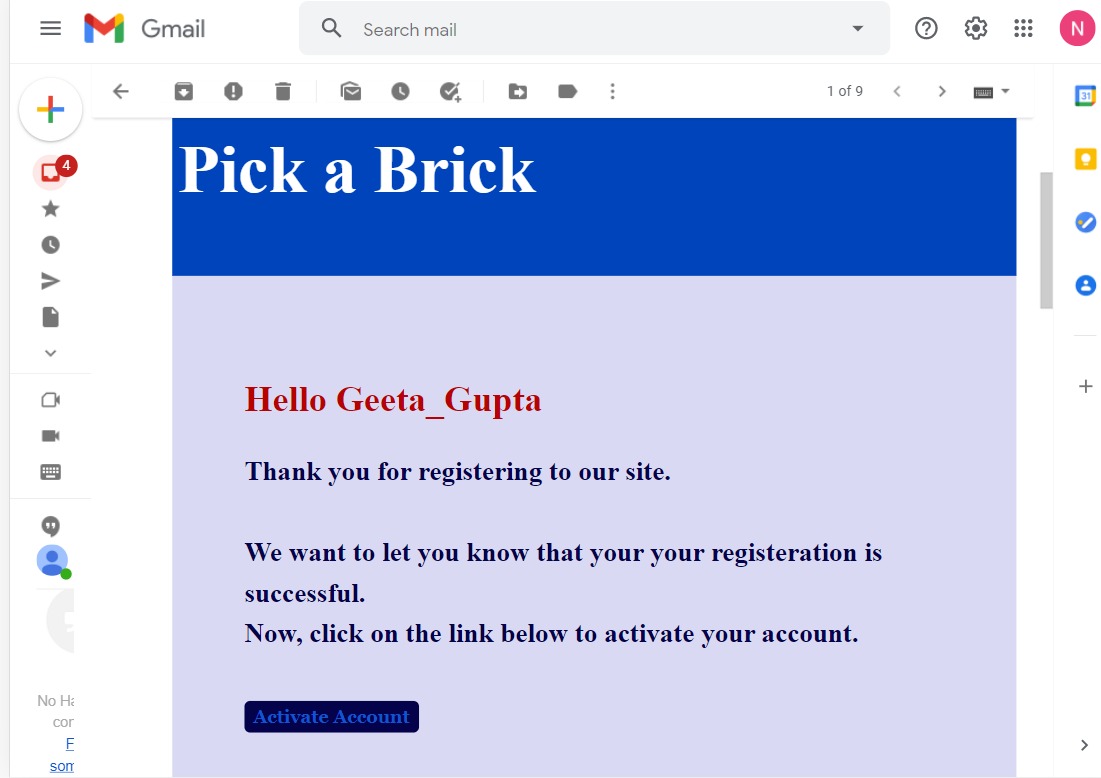
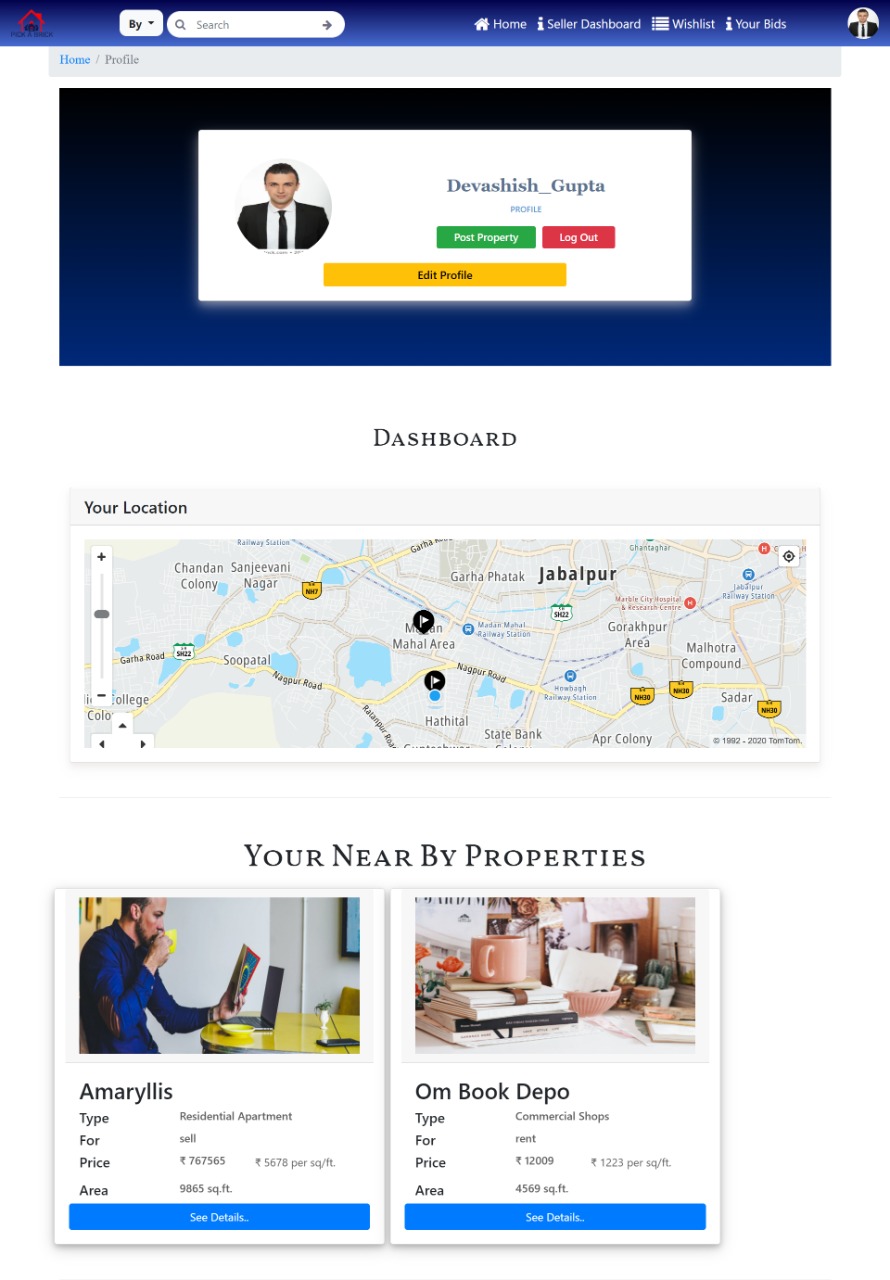
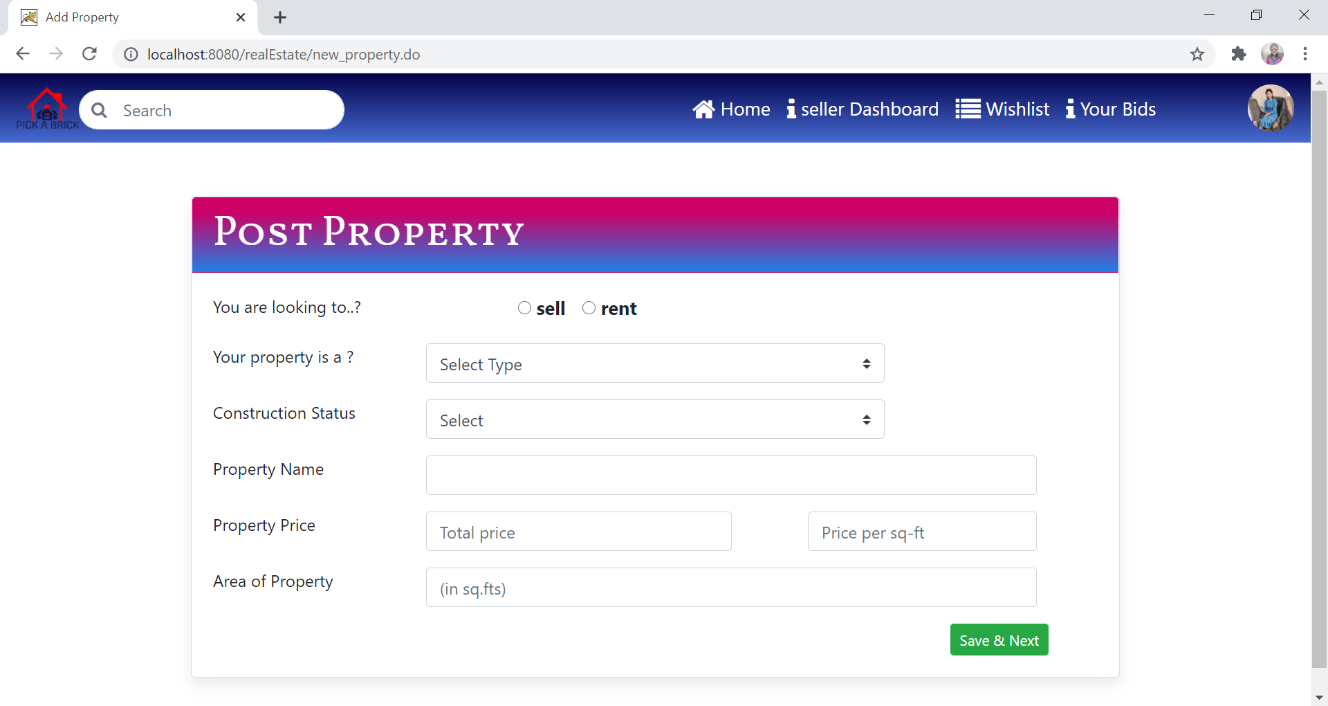
Chats Table

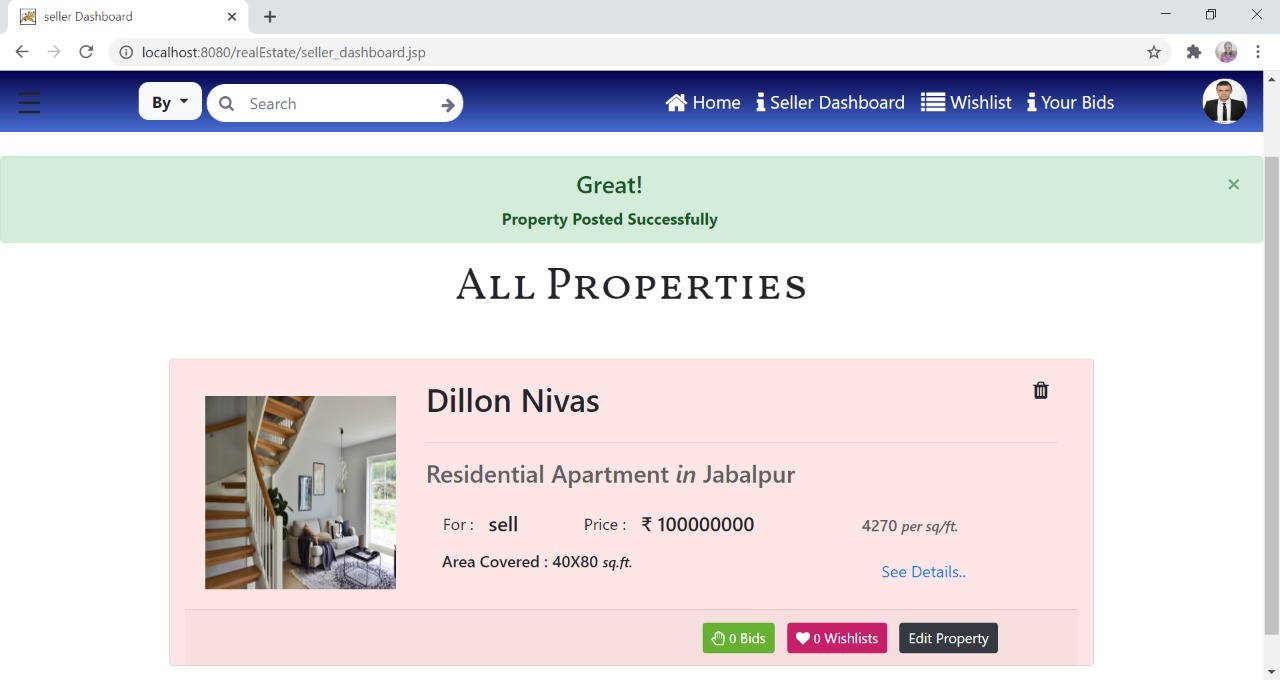


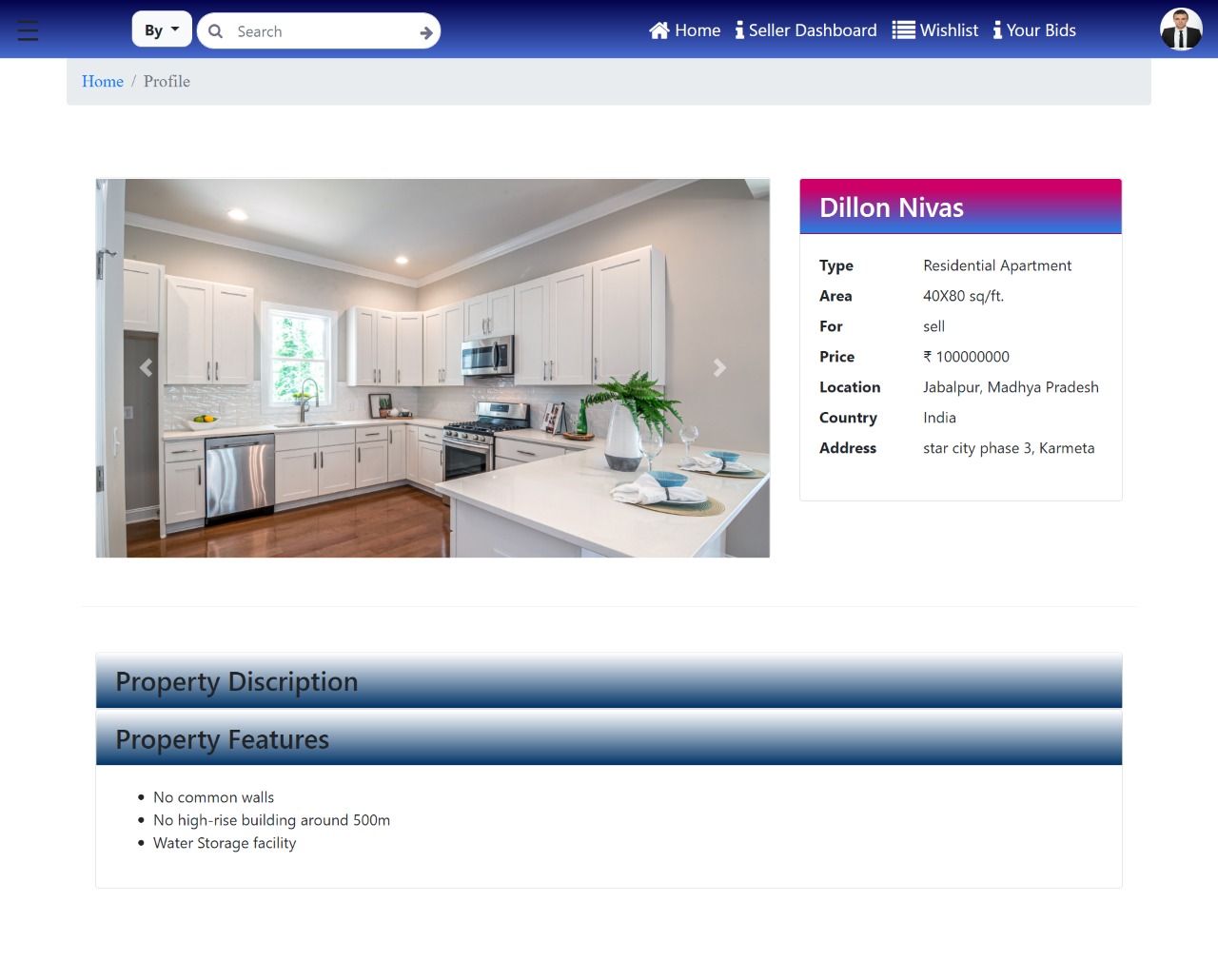
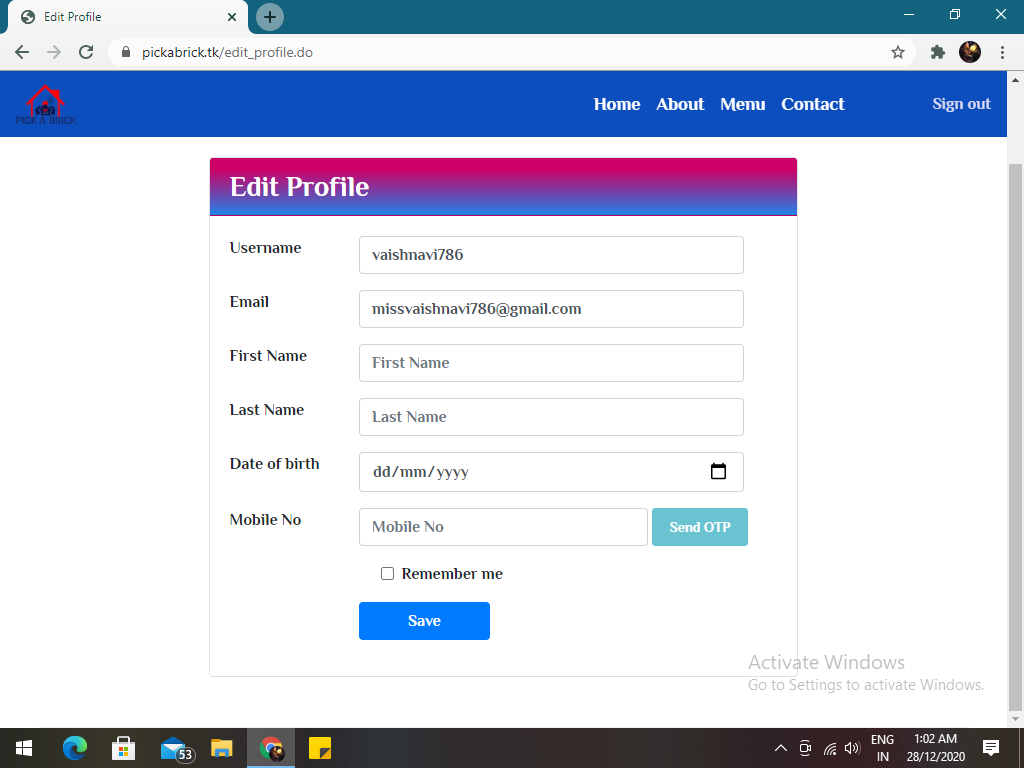
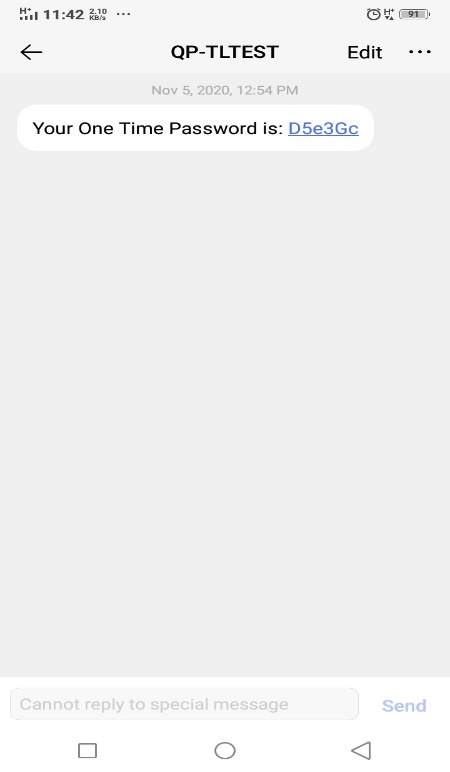
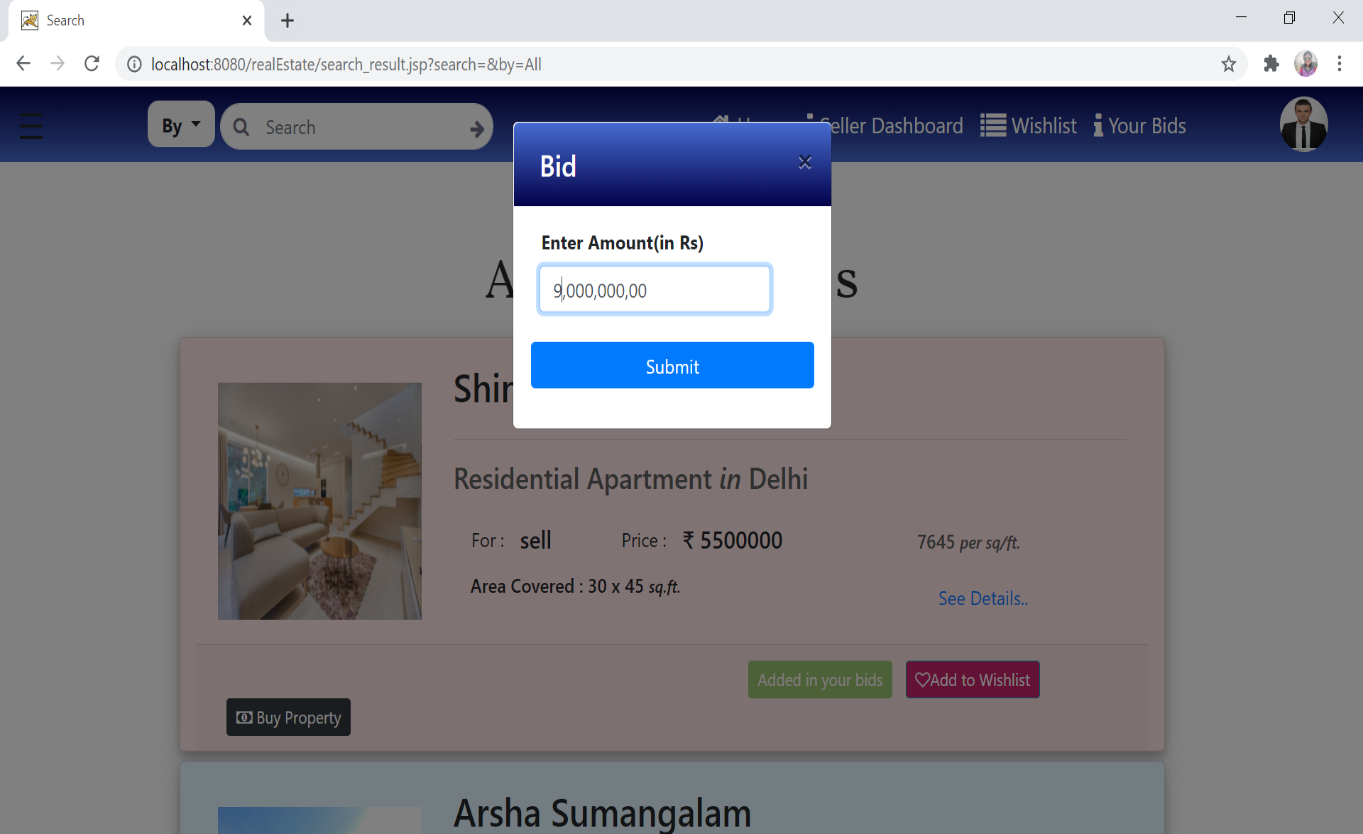
User\_Property Table:

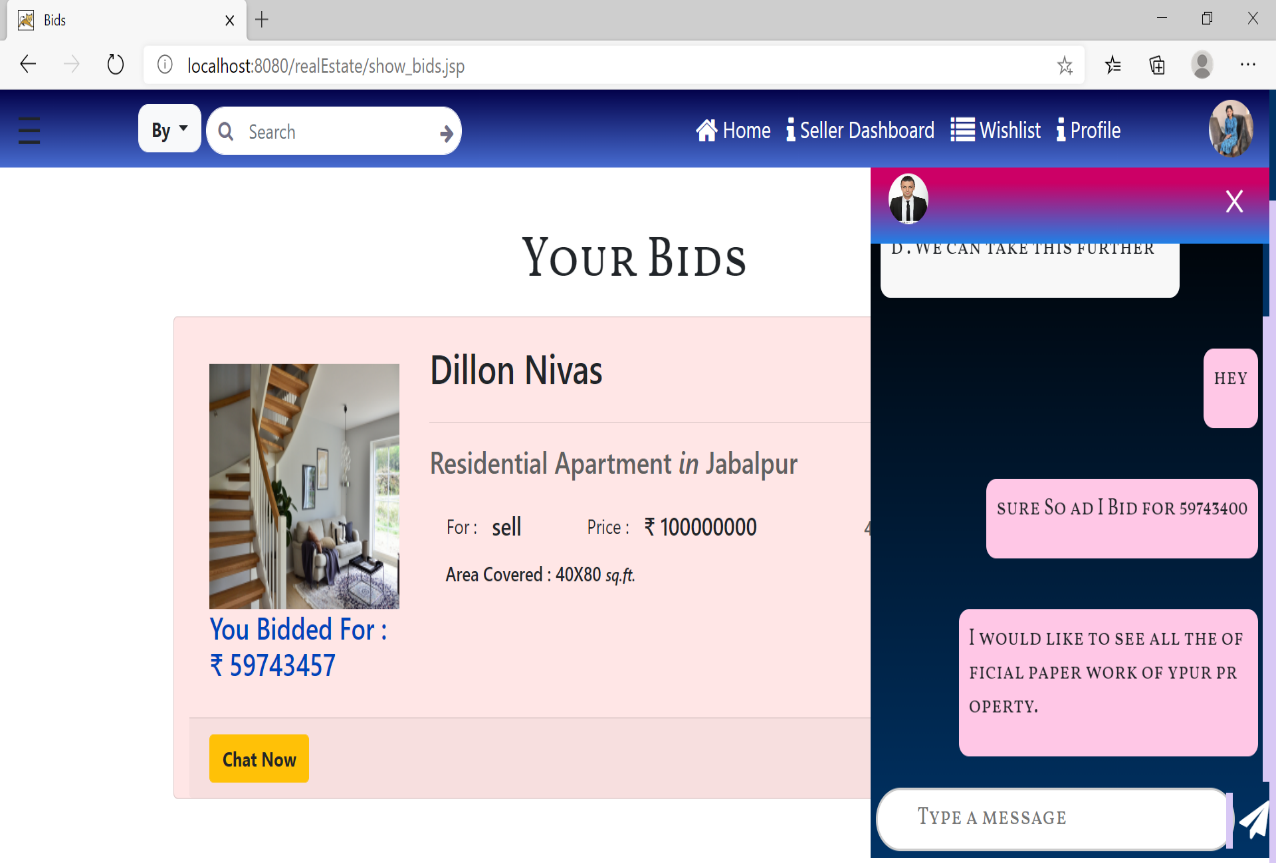


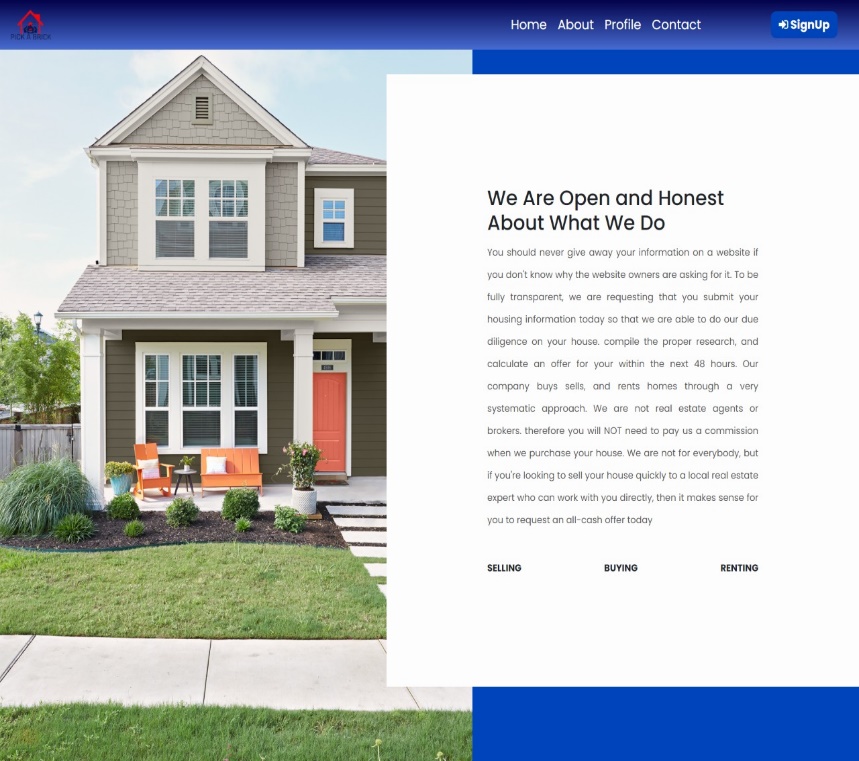
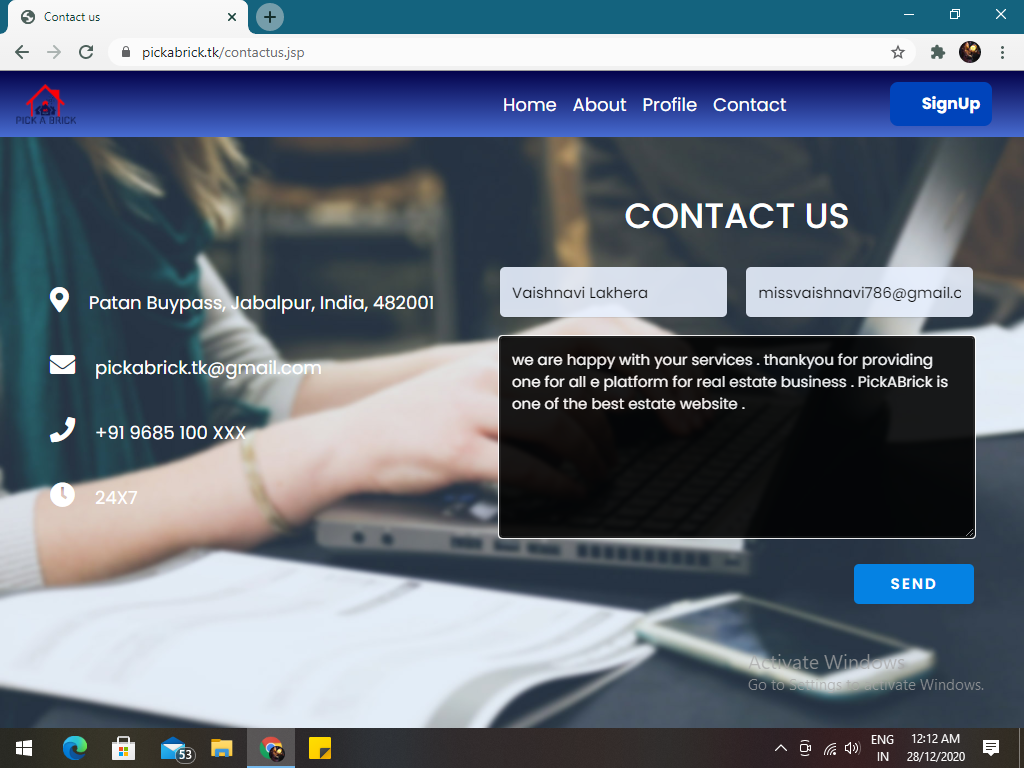
* System Design:
* Use Case Diagram:
* ****Data Flow Diagram:
* E-R Diagram:
* User Interface Design:
* Screen Images:

1.  Home Page:
2. Register/Login Page
3. Email Verification:
4. User Profile Page: Location is being tracked and thus providing nearby suggestion
5. Post Property:



1. View Property:
2. Edit Profile:
3. OTP Send:
4.  Bin on Property
5. Chat Now:



1. About US Page:
2. Contact us Page:

Conclusion

The development of the proposed system is done keeping in view the problems of agent i.e property dealer in the official dealings of properties. Typically, a dealer buys the property from one party and sells to another. Thus, the dealer’s profit is the difference between the price he pays to one party and the price he receives from another party for the same property. In short, the dealers have a particular commission in it.

Also, there are Real estate dealers who undertake a diversity of activities related to the selling and leasing of residential and commercial property. Many agents specialize in areas of practice such as residential sales, residential property management, commercial and industrial leasing and sales, representing buyers, rural sales and more.

For this we have to pay them with heavy amounts. With the use of this software, the property dealer does not need to store the records of dealings into registers or papers. So, it becomes easier for him to keep the records for future references. Also, it becomes much easier to keep a record of properties which are being sold or which are not sold yet. Thus, this system saves time and is easier to use. Moreover, it is efficient software for property dealers.

Future Scope:

* Verifying the authenticity of the property.
* At present one is unable to hire an agent to purchase property.
* Payment gateway.
* Subscription plans: to show a property in top ten list as well as in the home page, in the header of website

References

* <https://www.w3schools.com/bootstrap4/>
* w3schools.com/colors/colors\_gradient.asp
* <https://fonts.google.com/>
* <https://getbootstrap.com/docs/4.0/getting-started/>
* <https://docs.oracle.com/javase/10/docs/api/overview-summary.html>
* <https://docs.oracle.com/javaee/7/api/javax/servlet/package-summary.html>
* <https://developer.tomtom.com/>
* https://aws.amazon.com/