DIP NOTES

A project report submitted to

Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal

in partial fulfillment for the award of

the degree of

Bachelor of Technology

in

Information Technology

DEPARTMENT OF INFORMATION TECHNOLOGY

SUSHILA DEVI BANSAL COLLEGE OF TECHNOLOGY INDORE- 453331

2024 - 2025

DIP NOTES

A project report submitted to

Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal

in partial fulfillment for the award of

the degree of

Bachelor of Technology

in

Information Technology

PROJECT GUIDE
Prof. Nidhi Bhandari

SUBMITTED BY
Lalit Doriya (0829IT211035)
Devendra Prajapat (0829IT211023)
Dipanshu Panwar (0829IT211024)

DEPARTMENT OF INFORMATION TECHNOLOGY

SUSHILA DEVI BANSAL COLLEGE OF TECHNOLOGY INDORE- 453331

2024 - 2025

ACKNOWLEDGEMENT

I want to express my deep gratitude to all the faculty members who generously shared their time and expertise, helping me gain a comprehensive understanding of the project details and shaping the correct vision for its implementation. A special thanks goes to Prof. Vaishnavi Bangar for her unwavering commitment, engaging discussions, and the immense support she provided throughout the project. Her dedication and guidance have been invaluable.

-Lalit Doriya (0829IT211035)
-Devendra Prajapat (0829IT211023)
-Dipanshu Panwar (0829IT211024)

Dip Notes

A project report submitted to

Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal

in partial fulfillment for the award of

the degree of

Bachelor of Technology

in

Information Technology

DEPARTMENT OF INFORMATION TECHNOLOGY

SUSHILA DEVI BANSAL COLLEGE OF TECHNOLOGY INDORE- 453331

2024 - 2025

SUSHILA DEVI BANSAL COLLEGE OF TECHNOLOGY INDORE, 453331

CERTIFICATE

This is to certify that **Lalit Doriya (0829IT211035), Devendra Prajapat (0829IT211023),**

Dipanshu Panwar(0829IT211024) have completed their project work, titled **“DIP NOTES”** as per

the syllabus and have submitted by satisfy.

Part of fulfillment towards the degree of **“BACHELOR OF TECHNOLOGY”** (**INFORMATION**

TECHNOLOGY) from **RAJIV GANDHI PROUDYOGIKI** VISHWAVIDHYALAYA**, BHOPAL.**

HEAD OF THE DEPARTMENT **PROJECT GUIDE**

DIRECTOR

SUSHILA DEVI BANSAL COLLEGE OF TECHNOLOGY INDORE, 453331

CERTIFICATE

This is to certify that **Lalit Doriya (0829IT211035), Devendra Prajapat (0829IT211023), Dipanshu Panwar9(0829IT211024)** have completed their project work, titled **"Dip Notes"** as per the syllabus and have submitted by satisfactory.

Part of fulfillment towards the degree of **"BACHELOR OF TECHNOLOGY"** (**INFORMATION TECHNOLOGY)** from **RAJIV GANDHI PROUDYOGIKI** VISHWAVIDHYALAYA**, BHOPAL.**

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ABSTRACT

The Real Estate Web Application is an interactive, effective and revenue-generating website designed for the Real Estate Industry. The main objective of this application is to help the Real Estate Company to display unlimited number of property listings on the website.

The primary focus is to get familiar to .NET framework and code with ASP.NET and C# .NET to provide a featured GUI which contains sophisticated search engine for buyers to search for property listings specific to their needs. The search engine not only provides an easy and convenient way to search for listings but also display the entire list of properties in a customized grid format. The buyer can then view the complete specification of each property listing with its features, description and photographs. The application also provides a drag and drop control to save a list of selected property listings while browsing other options on the Real Estate Website.

There are hundreds of Real Estate Websites on the World Wide Web but the intention of designing this application is to develop something new, innovative and efficient using latest technologies like AJAX, Java Script, etc which not only enhances the already existing search features available on the internet but also gets rid of their annoying and unessential features. The main emphasis lies in providing a user-friendly search engine for effectively showing the desired results on the GUI.

TABLE OF CONTENTS

INTRODUCTION

CHAPTER 1 - Introduction

The motivation behind developing the Dip Notes website stems from a desire to create a centralized, user-friendly platform where B. Tech students from all branches and years can easily access academic notes and study material. The aim is to simplify the process of finding reliable notes by organizing them in a well-structured manner with efficient search and filter functionality.
This project serves as an opportunity to enhance my technical skills by building a full-stack web application using HTML, CSS, JavaScript for the frontend and backend technologies like Node.js and MongoDB for data storage and retrieval. Features like secure login/logout, responsive design, and dynamic content display will be implemented to provide a seamless user experience.
Another driving factor is the need for a platform that allows users to upload, access, and manage notes efficiently. The growing demand among students for easily available, branch-wise and semester-wise notes justifies the need for such a system. The project will also focus on providing fast access, search optimization, and user personalization, ensuring the platform remains useful and scalable for future enhancements.

1.2 Project Overview

1.2.1 Project Introduction

In today's digital age, students often struggle to find reliable and organized academic resources tailored to their curriculum. **Dip Notes** is a web-based platform designed to provide easy and structured access to academic notes for B. Tech students across all branches and semesters. The primary objective of this project is to offer a centralized repository where students can browse, download, and upload notes according to their academic requirements.
The platform categorizes notes based on branch, semester, and subject to simplify the search and selection process. This ensures that users can quickly locate the material most relevant to their needs. In addition, the website will incorporate features like user authentication, upload moderation, and responsive design to make the experience secure and smooth across different devices.
Dip Notes aims to build a collaborative academic community where students and educators can contribute and access high-quality content, ultimately promoting peer-to-peer learning and academic success.

1.1 **Objectives**

The primary objectives of the **Dip Notes** platform are as follows:

- To provide a centralized and structured repository of academic notes for B.Tech students of all branches and semesters.

- To simplify access to study materials through an intuitive and user-friendly web interface.
- To automate the process of uploading, organizing, and retrieving notes efficiently.
- To ensure accuracy, accessibility, and up-to-date information through community-driven content sharing.
- To support students' academic growth by enabling easy search, filter, and download functionalities.
- To incorporate user authentication and role-based access for secure uploading and downloading of materials.
- To offer responsive design, ensuring compatibility across devices including desktops, tablets, and smartphones.

In today's academic environment, students seek an efficient platform where they can quickly access relevant and high-quality notes. Dip Notes is built to address this need by implementing modern web technologies like JavaScript and AJAX to enhance interactivity without unnecessary page reloads.
Features like an advanced search tool, subject-wise categorization, and real-time suggestions make the platform robust and user-focused. Future enhancements may include drag-and-drop upload interfaces, personalized dashboards, and AI-based content recommendations to further enrich the user experience.

1.2.2 Problems with existing systems

The creation of the **Dip Notes** platform is driven by the need to address several shortcomings commonly observed in existing academic resource-sharing websites. Many current systems are either outdated, cluttered, or lack the user interactivity expected in modern web platforms. They often use static templates that simply display files without offering smart categorization, responsive design, or personalized search capabilities.

A major limitation is the **lack of structured navigation**—most platforms do not allow users to filter notes based on branch, semester, or subject, making it difficult for students to quickly find relevant materials. Additionally, some sites do not provide upload functionalities or moderation workflows, which leads to poor-quality or duplicate content.

Another key issue is the **inefficient user interface**, which relies on full-page reloads for every user action. This reduces user satisfaction and increases load times. By not utilizing modern technologies like AJAX or JavaScript-based dynamic rendering, these platforms fail to offer a smooth and interactive experience.

Moreover, many existing systems lack proper **admin control and data validation mechanisms**, leading to outdated, irrelevant, or unverified content. This compromises the reliability and usefulness of the platform.

Dip Notes seeks to resolve these challenges by offering:

- A responsive, AJAX-enabled interface that updates content dynamically without full page reloads.
- A powerful search engine with advanced filters.
- Secure, role-based access for uploading and moderating academic content.
- Clean, branch-wise and semester-wise categorization for efficient navigation.

These improvements ensure data integrity, improve usability, and provide a seamless academic resource-sharing experience.

1.2.3 **Objective**

1.4 Key Objectives and Innovations

The primary objective of the **Dip Notes** platform is to offer a **modern and engaging visualization** for an academic notes-sharing website—distinct from conventional, text-heavy educational portals. Most existing platforms follow a traditional layout where users must scroll through long lists or use basic dropdowns to find notes. In contrast, **Dip Notes introduces a dynamic and visually enhanced interface**, powered by AJAX and JavaScript, with features like collapsible menus, sliders, and responsive filtering components to improve the user experience.

A core technical objective is to **minimize full-page reloads** and avoid loss of user context using AJAX-enabled components. This ensures that users can search and filter notes across branches, semesters, and subjects without waiting for pages to reload, making the platform feel faster and more responsive.

To enhance interactivity, the platform also aims to provide:

- A **sliding bar or dropdown interface** for filtering notes by semester or difficulty level.
- **Accordion panels** to neatly organize subjects within each semester.
- A **drag-and-drop-style 'bookmark' feature** that allows users to temporarily save notes during a session—without the need to log in. This enables quick reference and comparison across multiple resources.

Unlike many academic portals that require mandatory sign-ins to access or bookmark content, **Dip Notes allows session-based saving of selected notes**, enabling a frictionless and user-friendly experience. This encourages engagement and accessibility, especially for students who need quick access without procedural delays.

By integrating such innovations, **Dip Notes** aims to become a feature-rich, efficient, and enjoyable platform for academic collaboration and learning.

.

1.2.4 Architecture

The **Dip Notes** web application is designed using a **modular 3-tier architecture**, which promotes separation of concerns and ensures scalability, maintainability, and performance. The architecture consists of:

- **Presentation Layer**: Built with HTML, CSS, and JavaScript (AJAX-enabled), this layer handles user interaction, providing a responsive and dynamic user interface. It includes features like semester filters, note categories, and drag-and-drop saving for personalized access.
- **Business Logic Layer**: This layer processes user requests, applies filtering logic, and manages session-based data such as bookmarked notes. It acts as the intermediary between the frontend and backend systems.
- **Data Access Layer**: Responsible for securely interacting with the backend database (e.g., MySQL or MongoDB), this layer handles CRUD operations on user details, notes content, file metadata, and user activity logs.

1.3 Requirement Specification

1.3.1 Scope

The scope of the **Dip Notes** project is to build a centralized, user-friendly web platform for sharing and accessing academic notes for all B.Tech branches and semesters. The application aims to enable students to upload, view, and download semester-wise notes categorized by subject, providing a seamless and efficient study support system.

1.3.2 Goal

The goal is to understand and implement the complete **Software Development Life Cycle (SDLC)**, from requirement gathering and system design to development, testing, and deployment. This project also helps gain hands-on experience in:

- Full-stack web development using HTML, CSS, JavaScript, and Node.js or PHP.
- Backend data handling and optimization using databases like MongoDB or MySQL.
- Implementing dynamic features like AJAX, file handling, and session management.

1.3.3 Assumptions

- Users will have access to a stable internet connection.
- Users will access the application through modern browsers (Chrome, Firefox, Edge, Safari).
- Uploaded files will be academic in nature and free from harmful content.
- The platform will be accessed primarily by students, faculty, and academic contributors.

1.3.4 Environment

- **Frontend**: HTML, CSS, JavaScript, AJAX.
- **Backend**: Node.js with Express.js or PHP (depending on chosen stack).
- **Database**: MongoDB or MySQL.
- **Development Environment**: Visual Studio Code or any modern IDE.
- **Hosting**: Localhost during development; deployed using services like Render, Vercel, or Heroku in production.

2.1 Technology Framework (Optional if required)

If you're using Node.js:

Node.js + Express Framework: Node.js is a fast, lightweight JavaScript runtime ideal for building scalable network applications. Express.js simplifies server logic and RESTful API creation.

If you're using .NET:

.NET Framework (If applicable): A powerful development platform used for building web applications and services. It includes tools, runtime environments, and libraries for efficient application development.

CHAPTER 2 – Developer Platform

The Dip Notes Web Application is developed using the Node.js platform, which supports fast and scalable network applications. The backend uses Express.js, a minimal and flexible Node.js web application framework. For persistent storage, the system uses MongoDB, a NoSQL database that allows efficient handling of semi-structured academic content such as notes, subjects, tags, and user uploads. Development and testing were carried out in Visual Studio Code, a modern code editor supporting JavaScript and web technologies.

The project is designed to be cross-platform and responsive, ensuring a smooth experience on desktops and mobile devices alike.

CHAPTER 3 – Technologies

This chapter outlines the tools and technologies used in the development of the Dip Notes Web Application, along with their role and advantages in building an efficient, scalable academic content-sharing platform.

3.1 Tools and Technologies

The latest technologies used to build the application include:

- Node.js
- Express.js
- MongoDB
- Visual Studio Code
- AJAX
- HTML/CSS/JavaScript

3.1.1 Node.js

Node.js is a JavaScript runtime built on Chrome's V8 engine, enabling developers to build fast and scalable server-side applications. It is non-blocking and event-driven, making it ideal for I/O-heavy applications like Dip Notes, where users are continuously uploading and retrieving academic content.

Node.js allows developers to use JavaScript across the stack (frontend and backend), ensuring code reusability and faster development cycles.

3.1.2 Express.js

Express.js is a lightweight web framework for Node.js that simplifies the development of RESTful APIs. It handles routing, middleware, and HTTP requests efficiently. In Dip Notes, Express is used to manage note uploads, file downloads, subject filtering, and user authentication.

Its middleware architecture allows clean separation of concerns and smooth integration with

---

3.1.3 MongoDB

MongoDB is a document-oriented NoSQL database. It stores data in JSON-like BSON format, allowing flexible schemas. It is well-suited for Dip Notes because each note document may include various fields like subject, semester, uploader, tags, and file path.

MongoDB supports powerful indexing and search features, enabling users to retrieve notes efficiently

---

3.1.4 Visual Studio Code

Visual Studio Code is a powerful open-source code editor by Microsoft. It supports syntax highlighting, debugging, version control, and intelligent code completion for JavaScript, Node.js, HTML, CSS, and more.

It enhances developer productivity with built-in terminal, Git integration, and a rich extension ecosystem.

3.1.5 AJAX and JavaScript

AJAX (Asynchronous JavaScript and XML) is used to enhance the user experience by making asynchronous server calls without reloading the page. For instance, when users filter notes by semester or subject, AJAX ensures a seamless experience by dynamically updating the content without a full page refresh.

This asynchronous interaction reduces server load and improves responsiveness, making the app feel modern and interactive.

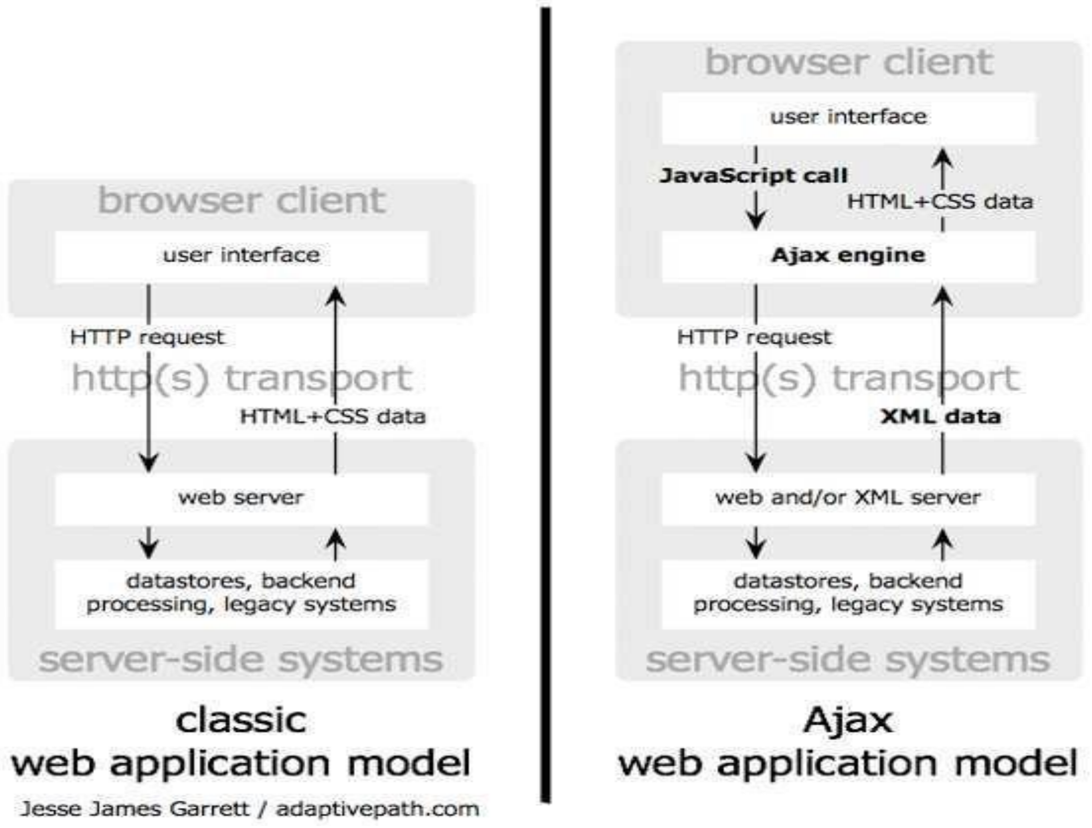


Figure 3.2 Comparisons of Classic Web Application Model and AJAX Web Application Model

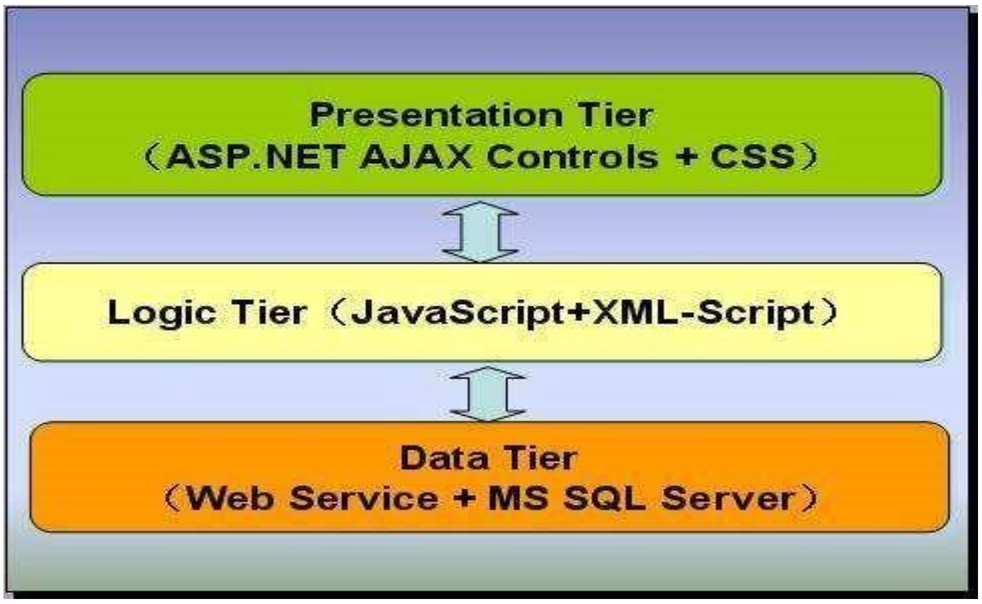
In many applications the sections of pages need to be reloaded rather than reloading the whole application, AJAX fulfills this requirement allowing for much more responsive web applications. Also, the use of Ajax can reduce connections to the server by its extensive use of since scripts and style sheets which have to be requested once.

4.1 System Architecture

The architecture of the system is based on the three-tier architecture. There are three logical layers: the Presentation Tier, the Business Tier, and the Data Tier. Each layer in the system is a reusable portion of code that performs a specific function. These layers are not only responsible for performing these functions but also interact with other layers to perform specific goals [4]. Like, in this application when the presentation layer needs to extract information from the backend database, it would utilize a series of layers to retrieve the data, rather than having the database calls embedded directly within it. The ASP.NET web site/web form is called the presentation layer because the content within it is viewable to the users of the site.

The middle tier called the Business Tier communicates between the presentation tier and the data tier. As mentioned above, the Presentation Layer could communicate with the data access layer directly, it usually goes through the business layer. This layer then validates the input conditions before calling a method from the data layer. This ensures the data input is correct before proceeding, and can often ensure that the outputs are correct as well. Most of the business logic of the application lies within the business layer, which makes this logic reusable across applications. This layer helps move logic to a central layer for "maximum reusability."

The Data Tier is the key component for this application as it is responsible for storing the data corresponding to each property listing. This layer is a separate component whose sole purpose is to serve up the data from the database and return it to the caller. Through this approach, data can be logically reused, meaning that a portion of an application reusing the same query can make a call to one data layer method, instead of embedding the query multiple times. This is generally more maintainable. Now this data is returned using the ADO.NET technology which is built into the .NET framework, ADO.NET contains a mechanism to query data out of the database and return it to the caller in a connected or disconnected fashion [5]. This scenario can be explained as, when the buyer makes a request to search for a particular property listing in a particular area (specified by the zip code/city/state/MLS#); the business layer passes this information to the database in the form of a query. The database retrieves the data corresponding to the query and submits the results back to the presentation tier for display. The Figure 4.1 shows the three-tier architecture of the Real Estate Web Application.


Presentation Tier
(ASP.NET AJAX Controls + CSS)
Logic Tier (JavaScript+XML-Script)
Data Tier
(Web Service + MS SQL Server)
Figure 4.1 3-Tier Architecture [10]

In the above figure, the Presentation Tier comprises of ASP.NET AJAX Controls which interact with the Logic Tier. The Presentation Layer can be elaborated to show what happens when the user requests something and calls to the server are made behind the scenes. This ASP.NET AJAX Framework uses the client-centric programming model. In this model, the large quantities of business logic from the server side are moved to the client side by introducing AJAX characteristics at the client-side controls.

This asynchronous communication is shown in the Figure 4.2:

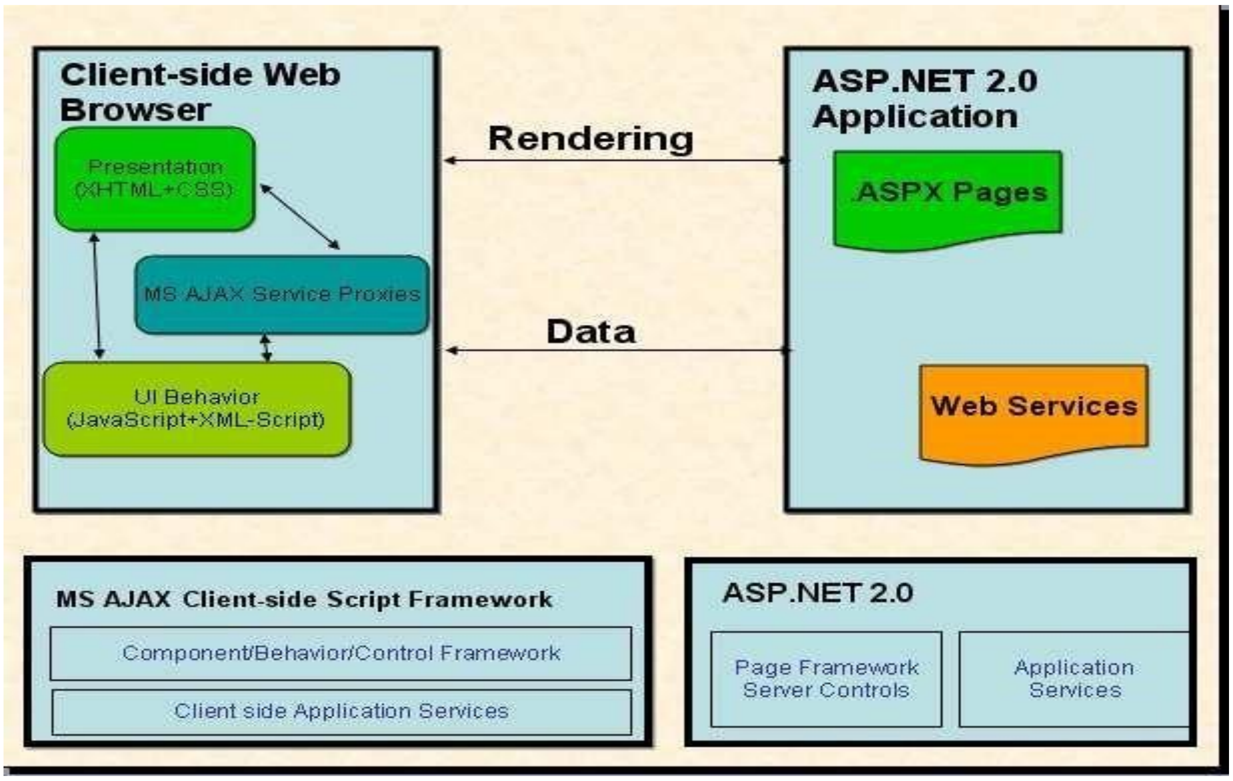

Client-side Web Browser
Presentation (XHTML+CSS)
MS AJAX Service Proxies
UI Behavior (JavaScript+XML-Script)
Rendering
Data
ASP.NET 2.0 Application
ASPX Pages
Web Services
MS AJAX Client-side Script Framework
Component/Behavior/Control Framework
Client side Application Services
ASP.NET 2.0
Page Framework Server Controls
Application Services

Figure 4.2 ASP.NET AJAX Frameworks

4.2 Architecture of Real Estate Web Application

The Real Estate Web Application is based on the three tier architecture namely the presentation tier, the middle tier, and the data tier.

. The application constitutes some of the basic pages needed by a Real Estate Application. This layer is responsible for the results/output from the business layer and transforms the results into a readable format for the end user. The following Page Flow Diagram presents the navigation flow in the application:

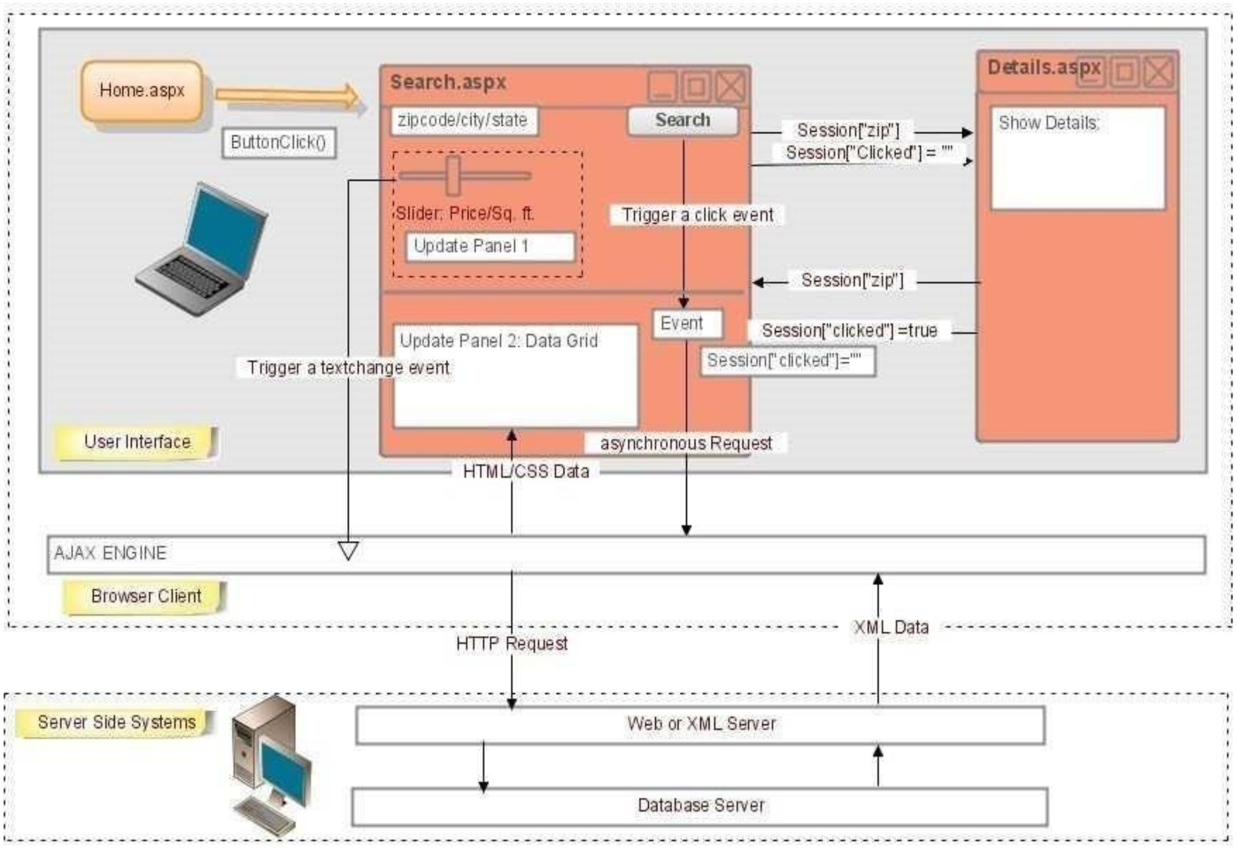


Figure 4.3 Page Flow Diagrams

Table 4.1 represents the list of ASP.NET Pages implemented in the Real Estate Web Application and the purpose of each page.

4.2.2.1 Use Case Diagram

Figure 4.4 shows the Use Case Diagram for the Real Estate Web Application. The buyer can search the property listings, view property details, enter the selected listings in the buyer cart and can delete the listings from the buyer cart.

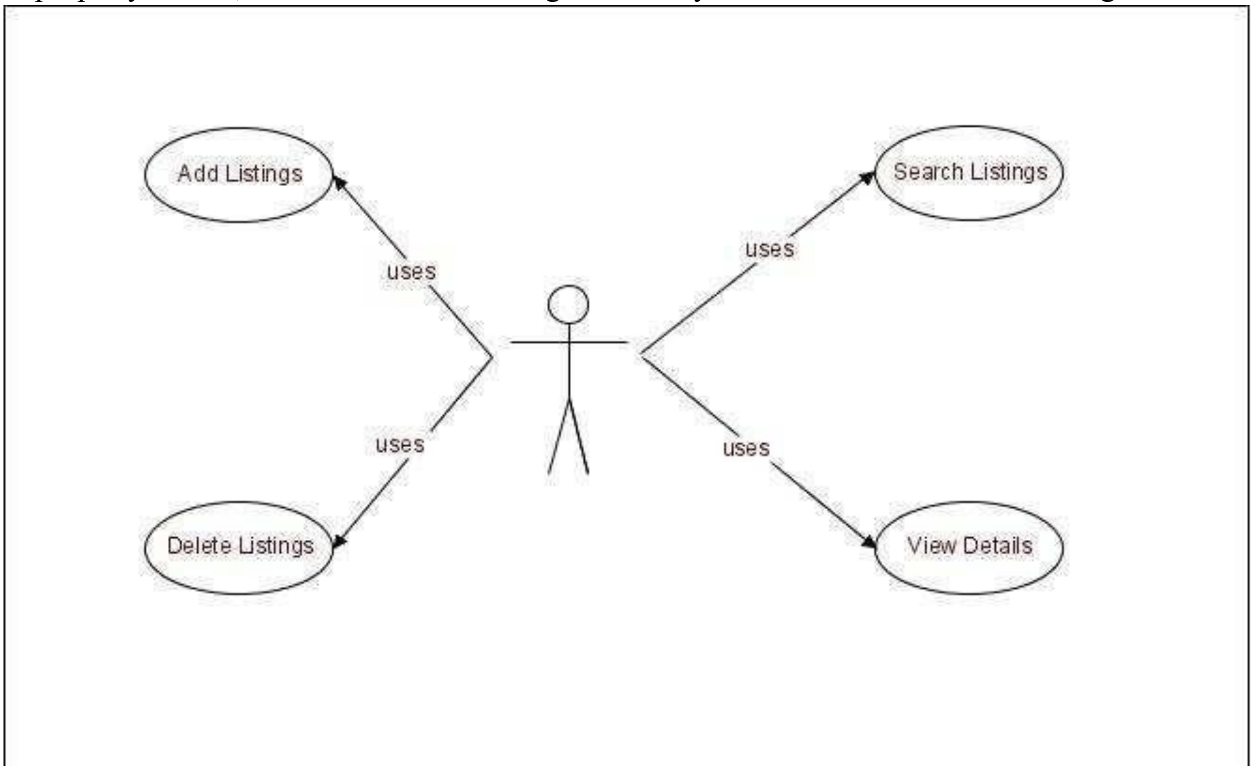


Figure 4.4 Use Case Diagram

4.2.2.2 Class Diagram

Figure 4.5 shows the Class Diagram representing the relations between the different classes.

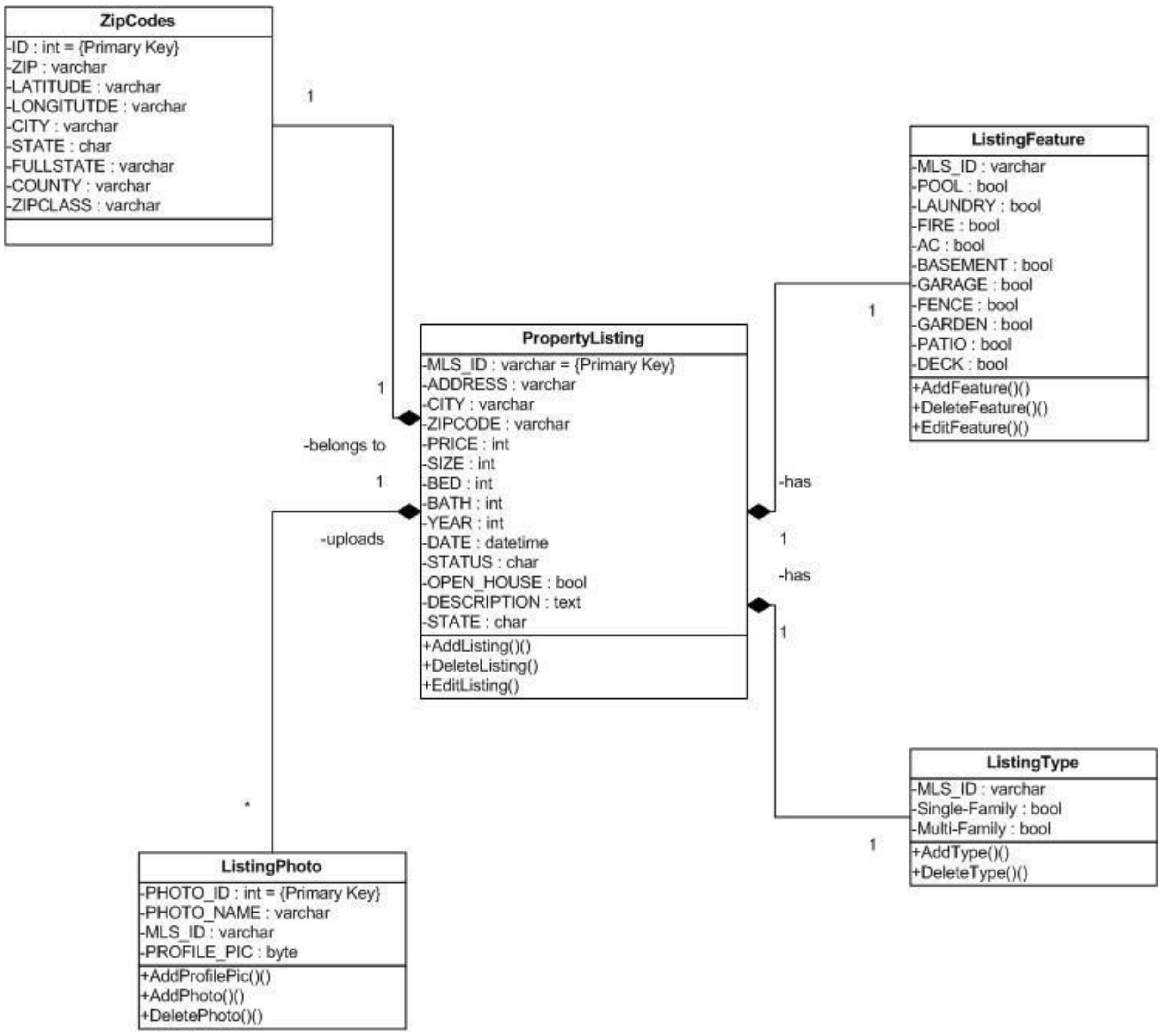


Figure 4.5 Class Diagram

The description for each of the classes in the Class Diagram is given below:

Property Listing Class:

The Property Listing class is the basic class for this application having attributes and description for each property listing which help the buyer to select the property desired for his need.

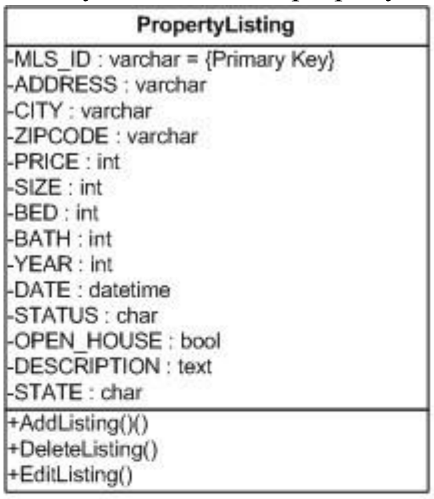


Figure 4.6 Property Listing Class

Listing Photo Class:

The Listing Photo class associates itself with the Property Listing. It is used to define the attributes of the photo that belongs to each Property Listing. The Real Estate Company can upload any number of photos but can have only one picture that acts as its profile picture. This profile picture of the listing is viewed whenever that particular listing pops up on the data grid by entering a particular search scenario. The rest of the pictures corresponding to the listing can be viewed in the photo album. The methods corresponding to this class are Add Photo and Delete Photo. The Class Diagram for Listing Photo is shown below:

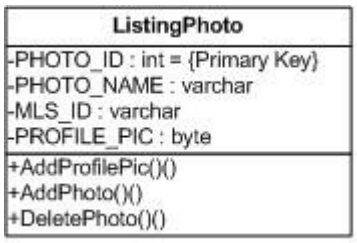


Figure 4.7 Listing Photo Class

4.2.3.1 ER Diagram

The Entity-Relationship model of the database being displayed is shown below:

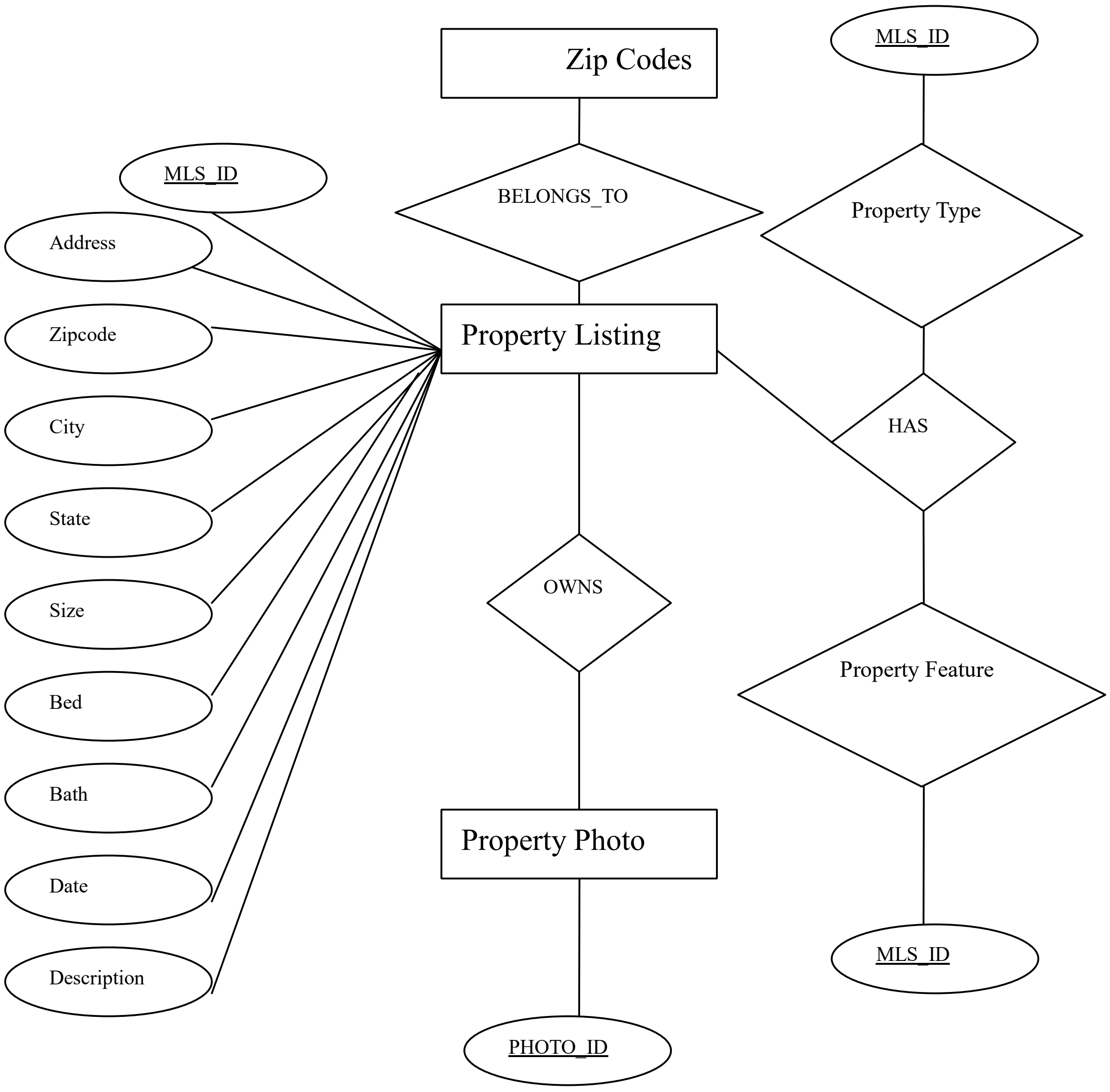


Figure 4.11 E-R Diagram

`

4.2.3.2 Database Schema

Following is the database schema diagram for the Real Estate Web Application, the associations between various database classes can be easily seen. The associations are also consistent with the class diagram's associations presented above. Member and Owns relationships are changed to strong entities in implemented.

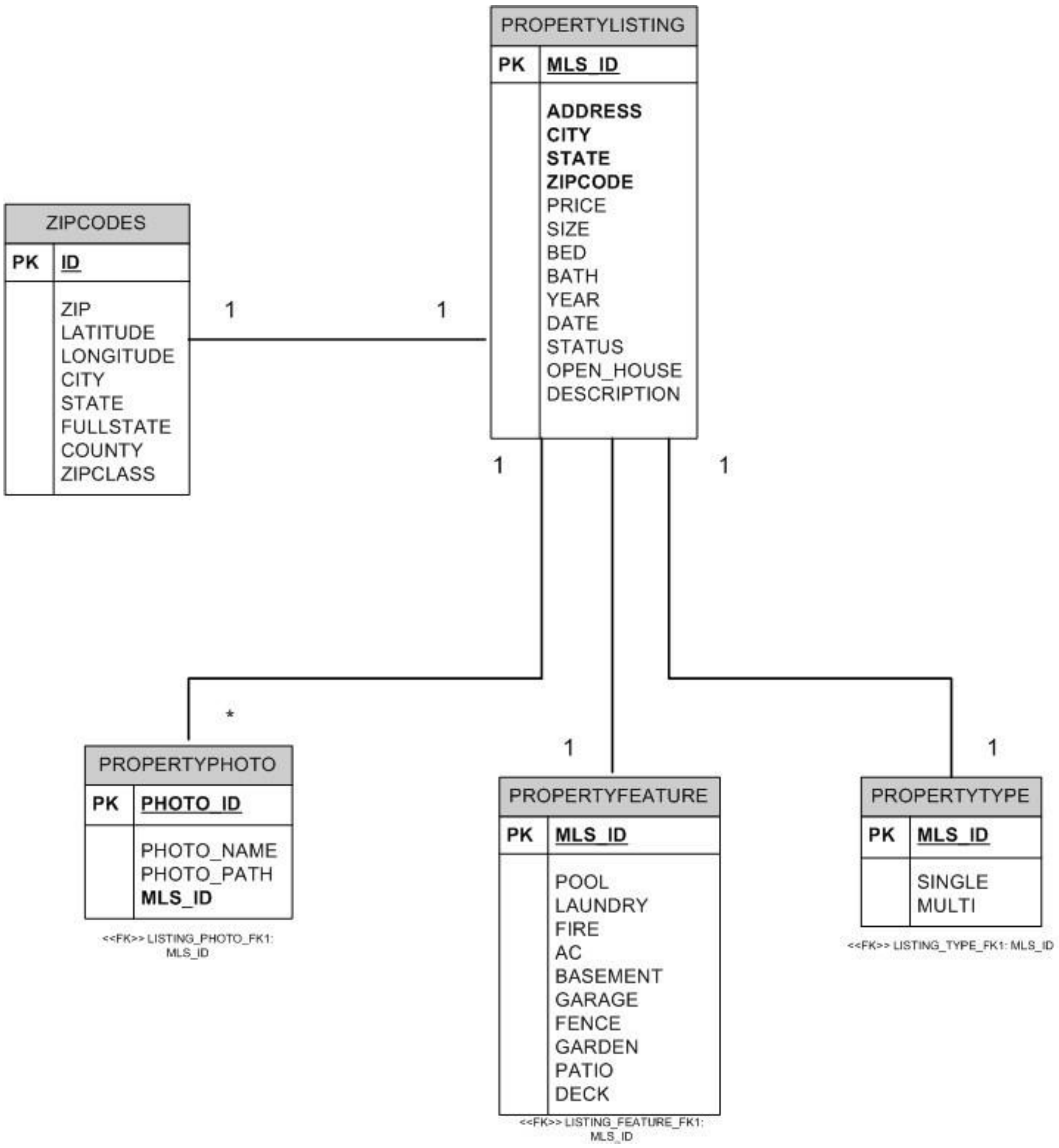
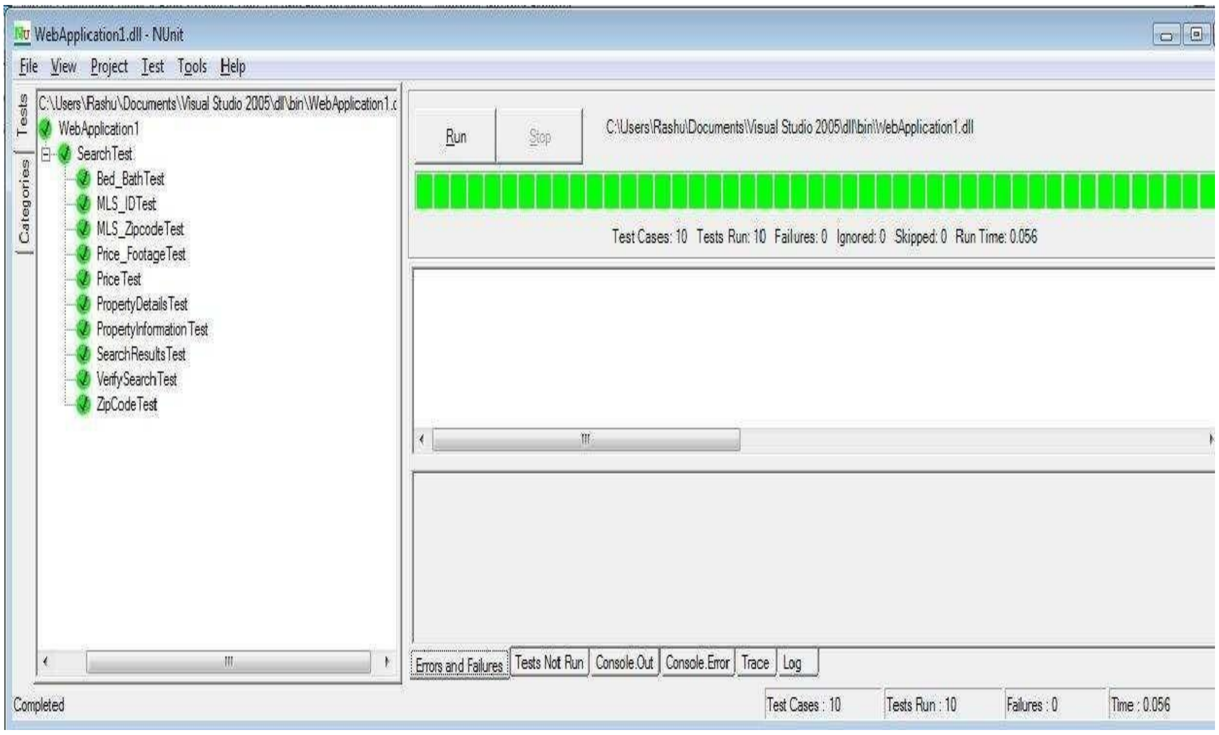


Figure 4.12 Database Schema

5.1 Unit Testing

The Real Estate Web Application uses NUnit Framework to perform Unit Testing. NUnit Framework is port of JUnit framework from java and Extreme Programming (XP) and is an open source product. This framework is developed to make use of .NET framework functionalities and uses an Attribute based programming model. The nunit.exe program is a graphical runner that shows the test cases in an explorer-like browser window and provides visual indication of success or failure of the tests.



```
C:\Users\Rashu\Documents\Visual Studio 2005\dll\bin\WebApplication1.c
WebApplication1
  SearchTest
    Bed_BathTest
    MLS_IDTest
    MLS_ZipcodeTest
    Price_FootageTest
    PriceTest
    PropertyDetailsTest
    PropertyInformationTest
    SearchResultsTest
    VerifySearchTest
    ZipCodeTest

C:\Users\Rashu\Documents\Visual Studio 2005\dll\bin\WebApplication1.dll
Test Cases: 10  Tests Run: 10  Failures: 0  Ignored: 0  Skipped: 0  Run Time: 0.056

Completed                    Test Cases : 10   Tests Run : 10   Failures : 0   Time : 0.056
```

Figure 5.1 Screenshot of NUnit test case execution.

5.2 Performance Testing

Performance of a web-application can be tested done by applying the Load and Stress Testing on it. Load testing is subjecting the application to a statistically representative load. This testing is done in support of the application reliability testing and in performance testing. In performance testing, the load is varied from minimum (zero) to the maximum level the application can handle without application-specific excessive delay.

Concurrent load [6].

Using the JMeter, I have performed several tests taking different readings and found out the peak number of users that can run effectively with this tool. The hardware capabilities of my machine limit the number of users to 1000. The testing results are shown in a tabular form containing the testing figures for various web pages. The testing is done keeping the loop count constant i.e. 16000 and Ramp-up period as 5 seconds and varying the number of users.

Users	Ramp-Up Period (sec)	Loop Count
100	5	16000
500	5	16000
1000	5	16000

Table 5.2: Test Cases Summary Testing is

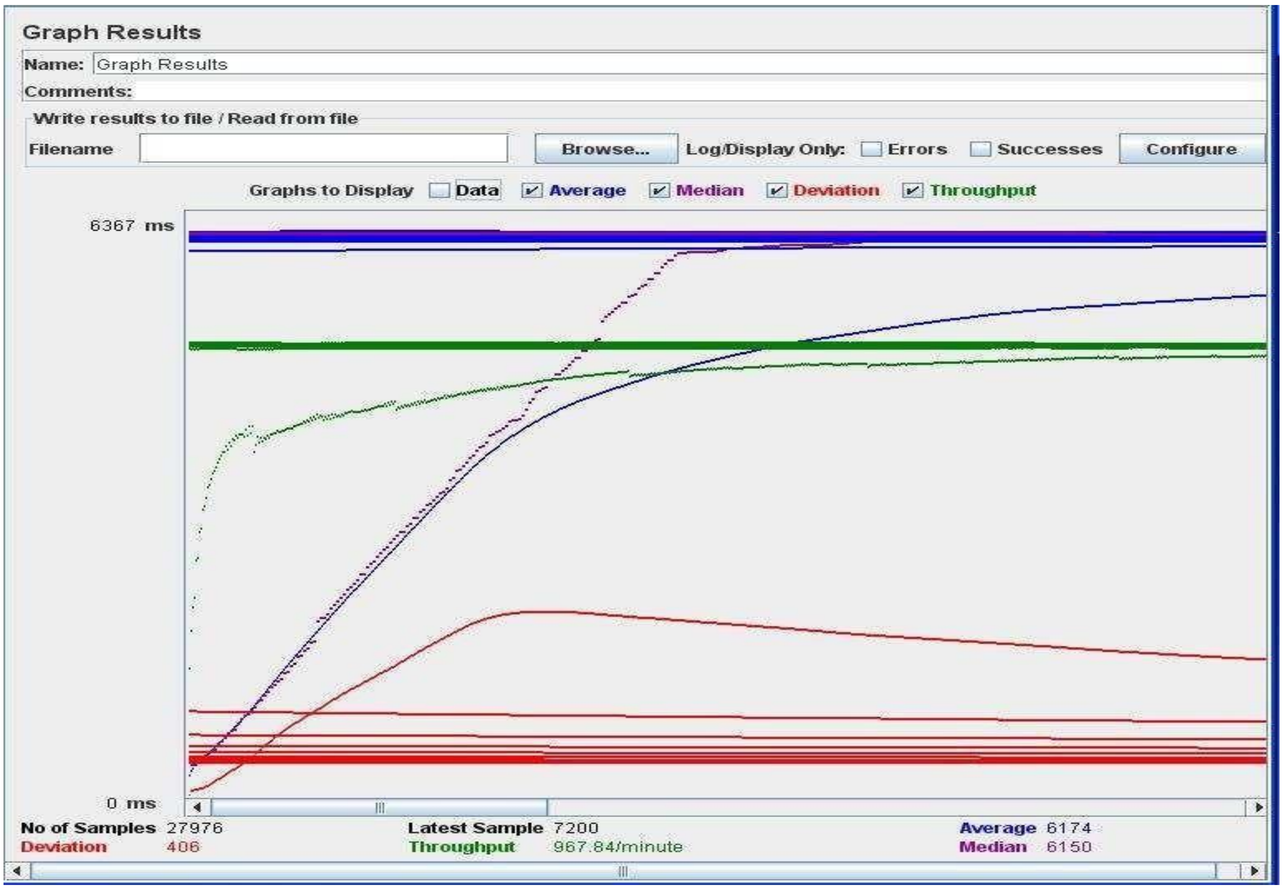
5.2 Testing Samples

5.2.1 Search Web Page

5.2.1.1 Test Case 1

Users	Loop Count	Ramp-up period (sec)	Average Response Time (ms)	Throughput
100	16000	5	6174	967.84/min

Table 5.3: Search Page Test Case 1



Graph Results

Name: Graph Results

Comments:

Write results to file / Read from file

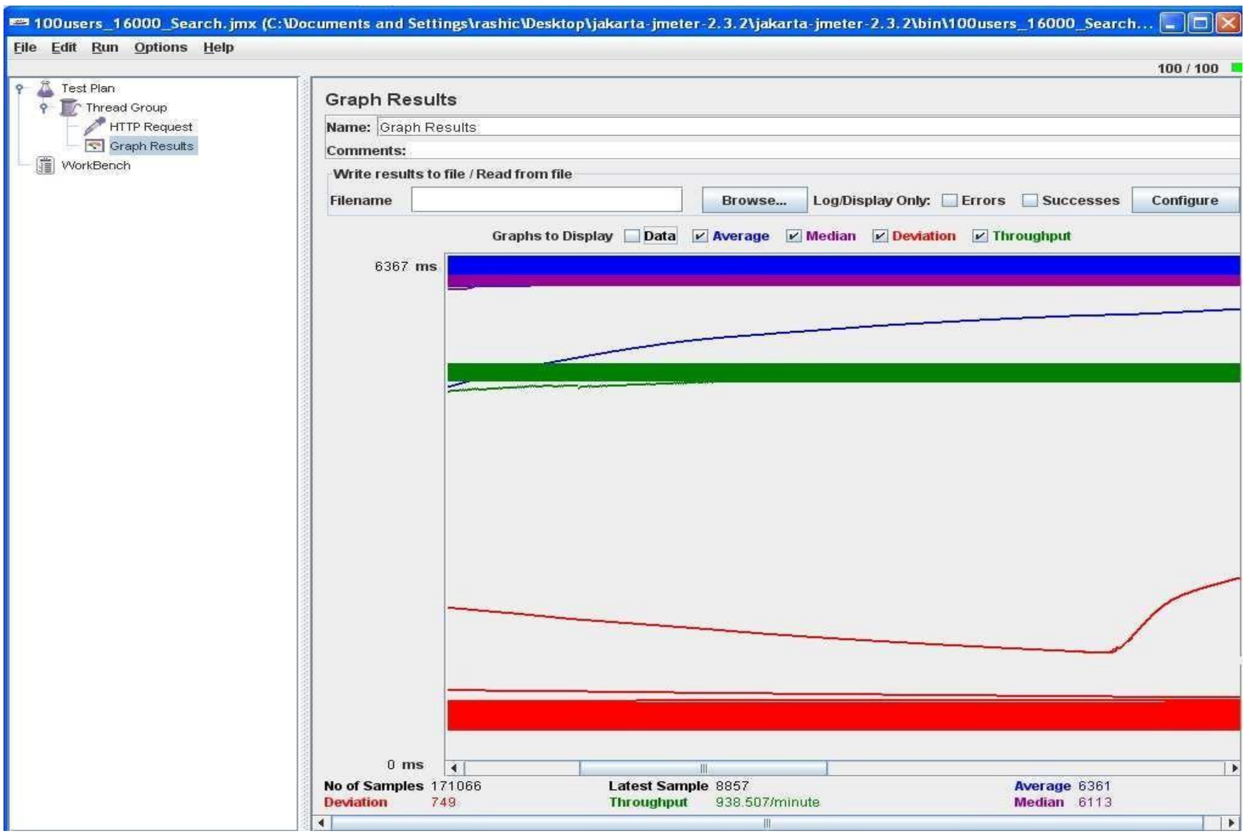
Filename

Log/Display Only: ☐ Errors ☐ Successes

Graphs to Display ☐ Data ✓ Average ✓ Median ✓ Deviation ✓ Throughput

No of Samples 27976
Deviation 406
Latest Sample 7200
Throughput 967.84/minute
Average 6174
Median 6150

Figure 5.2 Search Page Test Case 1.1 - Graph Results



100users_16000_Search.jmx (C:\Documents and Settings\rashic\Desktop\jakarta-jmeter-2.3.2\jakarta-jmeter-2.3.2\bin\100users_16000_Search...

Graph Results

Name: Graph Results

Comments:

Write results to file / Read from file

Filename

Log/Display Only: ☐ Errors ☐ Successes

Graphs to Display ☐ Data ✓ Average ✓ Median ✓ Deviation ✓ Throughput

No of Samples 171066
Deviation 749
Latest Sample 8857
Throughput 938.507/minute
Average 6361
Median 6113

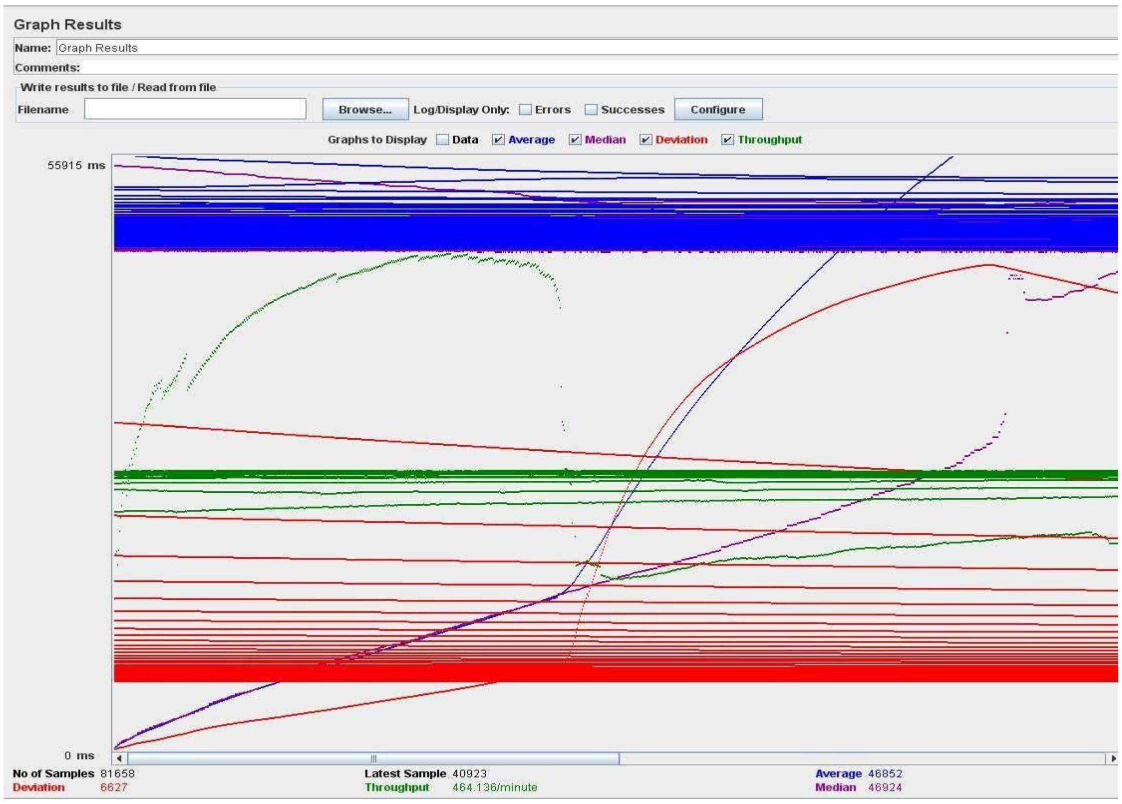
Figure 5.3 Search Page Test Case 1.2 - Graph Results

5.2.1.2 Test Case 2

Users		Loop Count	Ramp-up period (sec)	Average Response Time (ms)	Throughput
500		16000	5	55915	967.84/min

6

Table 5.4: Test Case 2


Graph Results

Name: Graph Results

Comments:

Write results to file / Read from file

Filename

Log/Display Only: ☐ Errors ☐ Successes

Graphs to Display ☐ Data ☑ Average ☑ Median ☑ Deviation ☑ Throughput

No of Samples 81658
Deviation 6627
Latest Sample 40923
Throughput 464.136/minute
Average 46852
Median 46924

Figure 5.4 Search Page Test Case 2 - Graph Results

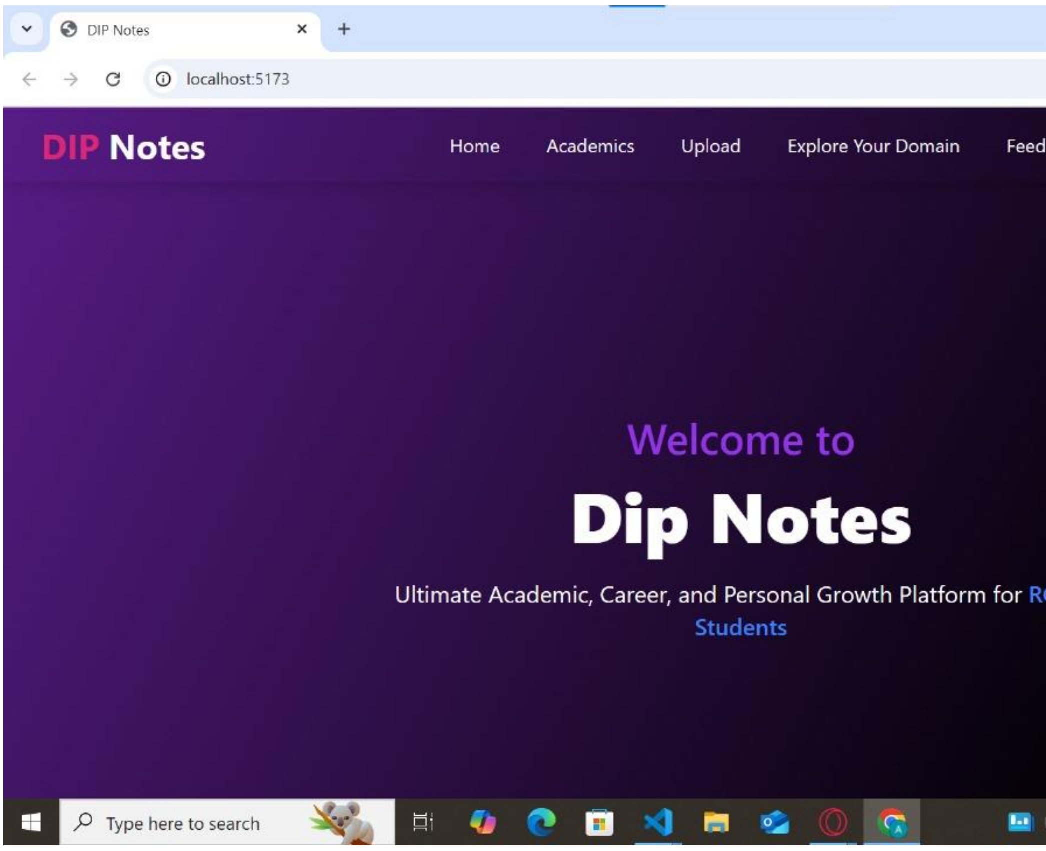
5.2.1.3 Test Case 3

Users	Loop Count	Ramp-up period (sec)	Average Response Time (ms)	Throughput
1000	16000	5	33243	364.157/min

Table 5.5: Search Page Test Case 3

5.3 Screen Shots of Tested Web Pages

5.3.1 about Us Webpage



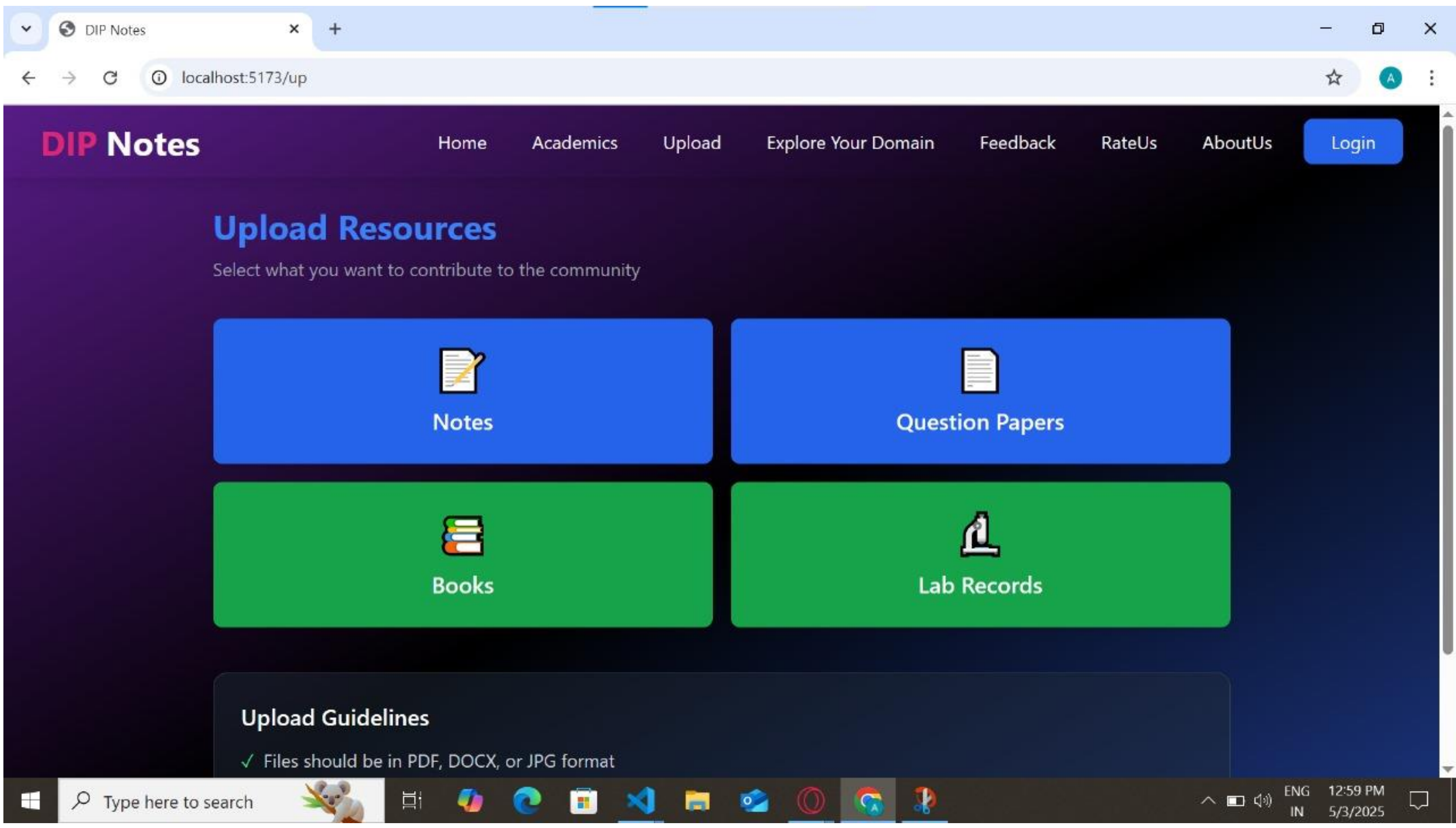
Welcome to

Dip Notes

Ultimate Academic, Career, and Personal Growth Platform for R
Students

Figure 5.10 Screenshot of About Us Webpage

5.3.2 Search Webpage



Upload Resources

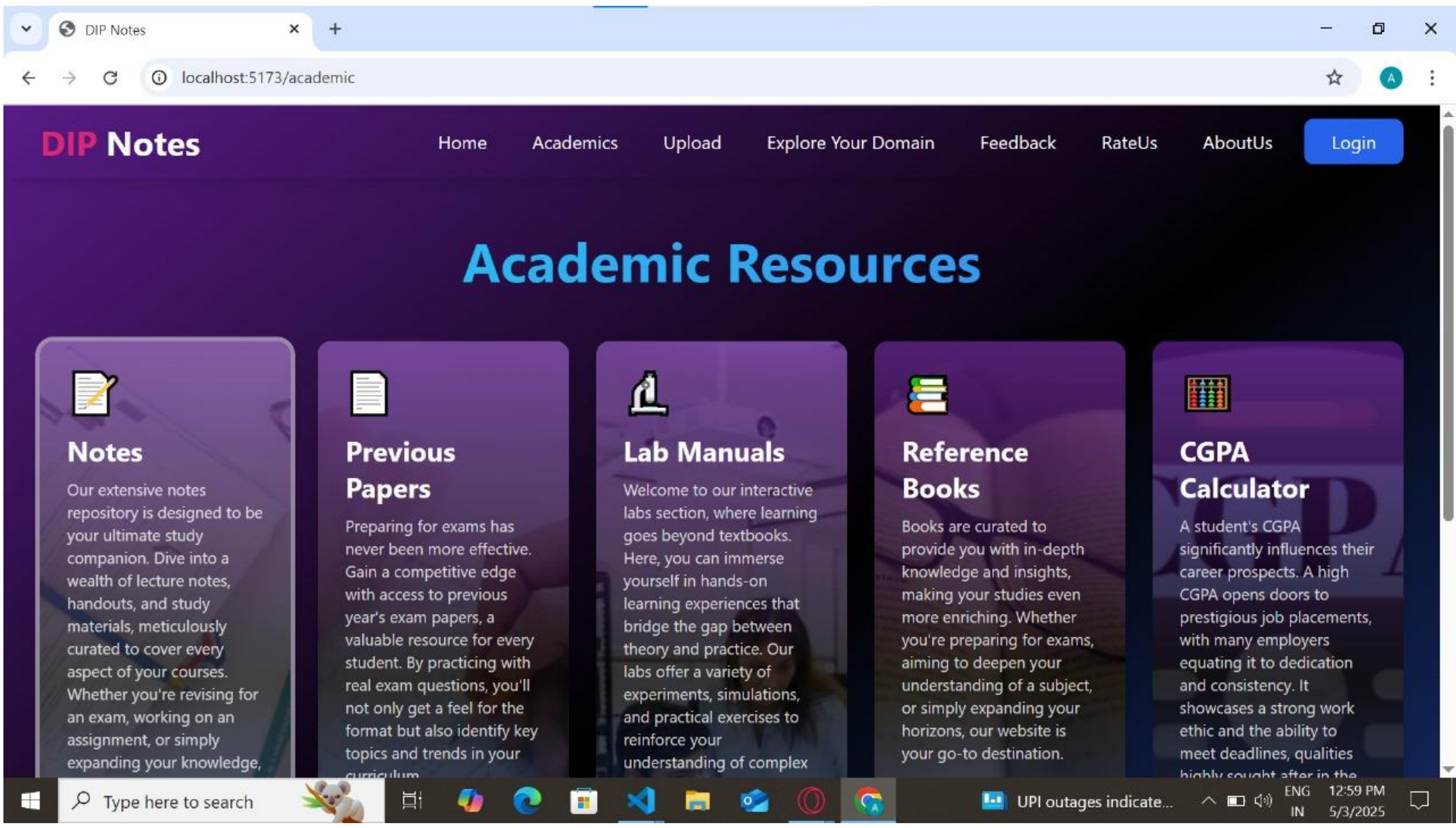
Select what you want to contribute to the community

Upload Guidelines

✓ Files should be in PDF, DOCX, or JPG format

Figure 5.11 Screenshot of Search Webpage

Figure 5.12 Screenshot of Search Webpage with Drag and Drop Tool



Academic Resources

Notes
Our extensive notes repository is designed to be your ultimate study companion. Dive into a wealth of lecture notes, handouts, and study materials, meticulously curated to cover every aspect of your courses. Whether you're revising for an exam, working on an assignment, or simply expanding your knowledge,

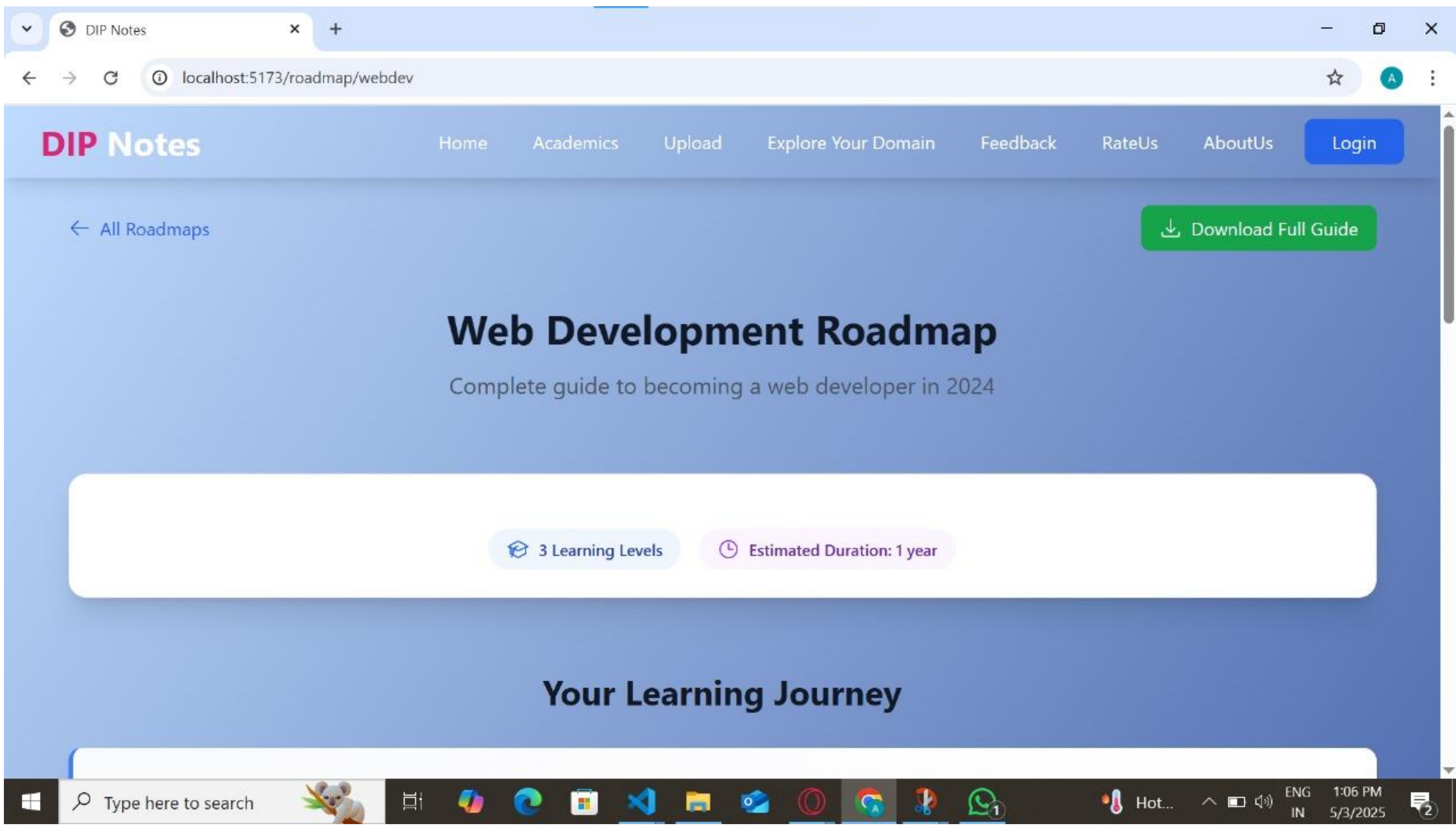
Previous Papers
Preparing for exams has never been more effective. Gain a competitive edge with access to previous year's exam papers, a valuable resource for every student. By practicing with real exam questions, you'll not only get a feel for the format but also identify key topics and trends in your curriculum

Lab Manuals
Welcome to our interactive labs section, where learning goes beyond textbooks. Here, you can immerse yourself in hands-on learning experiences that bridge the gap between theory and practice. Our labs offer a variety of experiments, simulations, and practical exercises to reinforce your understanding of complex

Reference Books
Books are curated to provide you with in-depth knowledge and insights, making your studies even more enriching. Whether you're preparing for exams, aiming to deepen your understanding of a subject, or simply expanding your horizons, our website is your go-to destination.

CGPA Calculator
A student's CGPA significantly influences their career prospects. A high CGPA opens doors to prestigious job placements, with many employers equating it to dedication and consistency. It showcases a strong work ethic and the ability to meet deadlines, qualities highly sought after in the

5.3.3 **Details Webpage**



Web Development Roadmap

Complete guide to becoming a web developer in 2024

Your Learning Journey

Figure 5.13 Screenshot of Details Webpage

CHAPTER 6 - Project Metrics and Experience

This chapter presents the project metrics showing the number of hours spent completing each phase of the project. It also summarizes the experiences gained during the entire life-cycle of the project.

6.1 Project Metrics

Project Metrics are the indicators that track the ongoing project progress. The Project Metrics discussed in this document are source lines of code and the amount of time spent during the entire project span. Table 1 and Table 2 represent the project metrics: source lines of code and project phases and their duration respectively.

ASP.NET & C# Server Side Code	Handwritten C# Code – 1180 lines approx. Auto-generated C# Code – 750 lines approx.
SQL Code	300 lines
CSS Code	500 lines

Table 6.1: Project Lines of Code

Learning Project Technologies	1 Week
Requirement Gathering and Design	2 Week
Implementation	4 Weeks
Testing	1 Week
Documentation	2 Weeks

Table 6.2: Project Planning Phase

6.2 Overall Experience

The idea of developing Real Estate Web Application originated when an immediate requirement aroused from local Real Estate Company to develop a new website for them. This company had its own website in production but wanted to re-design it with some attractive features which could help them fetch more business. The project started with an intention to develop an application specific to the needs of this company but later due to the lack of funds the project could not be continued. But with the continuous support of Dr. Andresen this application came into development. There are hundreds of real estate websites on the World Wide Web with same features but the intention of building this application was to design something new and innovative and include some cool features which have not been incorporated in these websites so far. The biggest challenge involved in this project was to gather the requirements and design its structure so that it can altogether have a new look. It also involved thinking about new features which can be incorporated in this application and could make the search of listings much easier for the real estate buyers. Understanding the structure of the application was the biggest problem on the start of the project. Finally, the scope of this application was defined which greatly helped in understanding what all features have to be included in the project. The whole emphasis in this application is given on the search criteria to help buyers to search for new property listings.

CHAPTER 7 - Conclusion and Future Work

This chapter describes the future scope and extensions for the project. There is still a huge scope of implementing something new and more to the project which can make it to the level of a commercial product. This section also concludes stating the advantages and applications of this Real Estate Web Application.

7.1 Conclusion

This Real Estate Web Application is a typical .NET web application using ASP.NET 2.0 and SQL 20005 in the C# programming language. It uses a client/server architecture based on the HTTP protocol. It is developed in Microsoft's Visual Studio .NET programming environment. The buyer performs a search for the property listings by putting either Zip code/City/State or MLS# in the search textbox. The business logic tier communicates with the database tier requesting the results of the query sent by it. The results obtained by the database are displayed on the data grid, by refreshing the grid rather than refreshing the entire web page. Efficiency of the application is improved by the use of web methods that help in separating Application Tier from the Presentation Tier. The performance of this application is evaluated by rigorously testing it against various test scenarios. Efficiency and correctness of the application is evaluated with the help of various test cases. Some ways in which this system could be enhanced with additional functionalities are discussed in the section.

7.2 Using Object-Oriented Domain Model

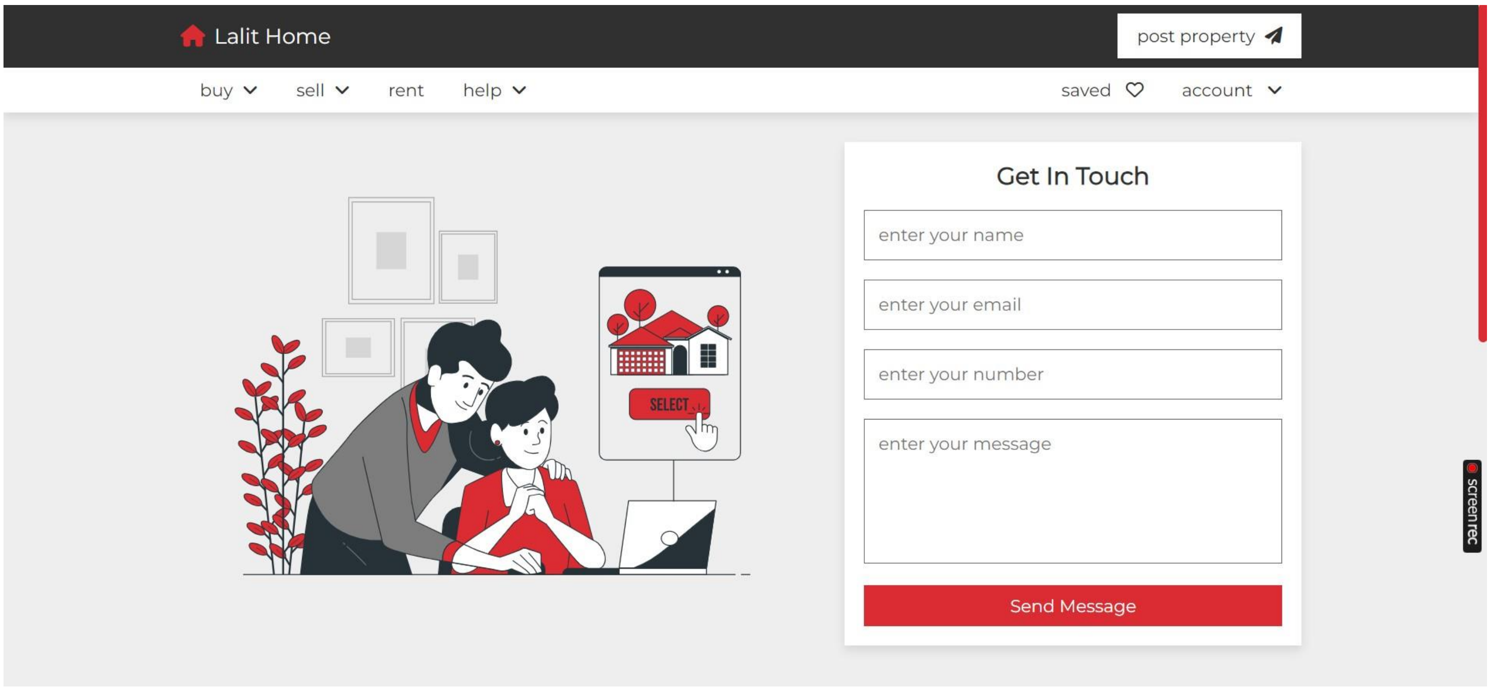
Visual Studio .NET 2005 and SQL Server 2005 add a significant feature of object-oriented domain model. Instead of using traditional Relational Database Model, the database access layer treat each table as class model and each row as instance object. It gives a friendly development thought to the developer and makes the business logic more convenient when interacting with the back end. Currently, there is a thirdparty tool called NEO which could generate the domain model for database automatically. In using objectoriented domain model, the database is transported to the developer and the application using these domain classes to handle data processing instead of using disconnected Dataset or typed Dataset.

7.2.1 Functionality Extensions

If we are going to develop commercial real estate web application based on this project, we can add some more feature as the following:

1. We can provide a login page for buyers and store their username and password in the database and can save the search criteria for the buyers. This will help the buyers to use the same search criteria rather than creating a new criterion every time the buyer performs the search.
2. By using the AdRotator web control we could add more advertisement on the web site or links to other web site.
3. Add agent and company search functionality.
4. Displaying the search results on Google Maps locating the area where the property listing is located.

5. Refine this web site and make it friendly and pretty.


Lalit Home

post property

buy sell rent help

saved account

Get In Touch

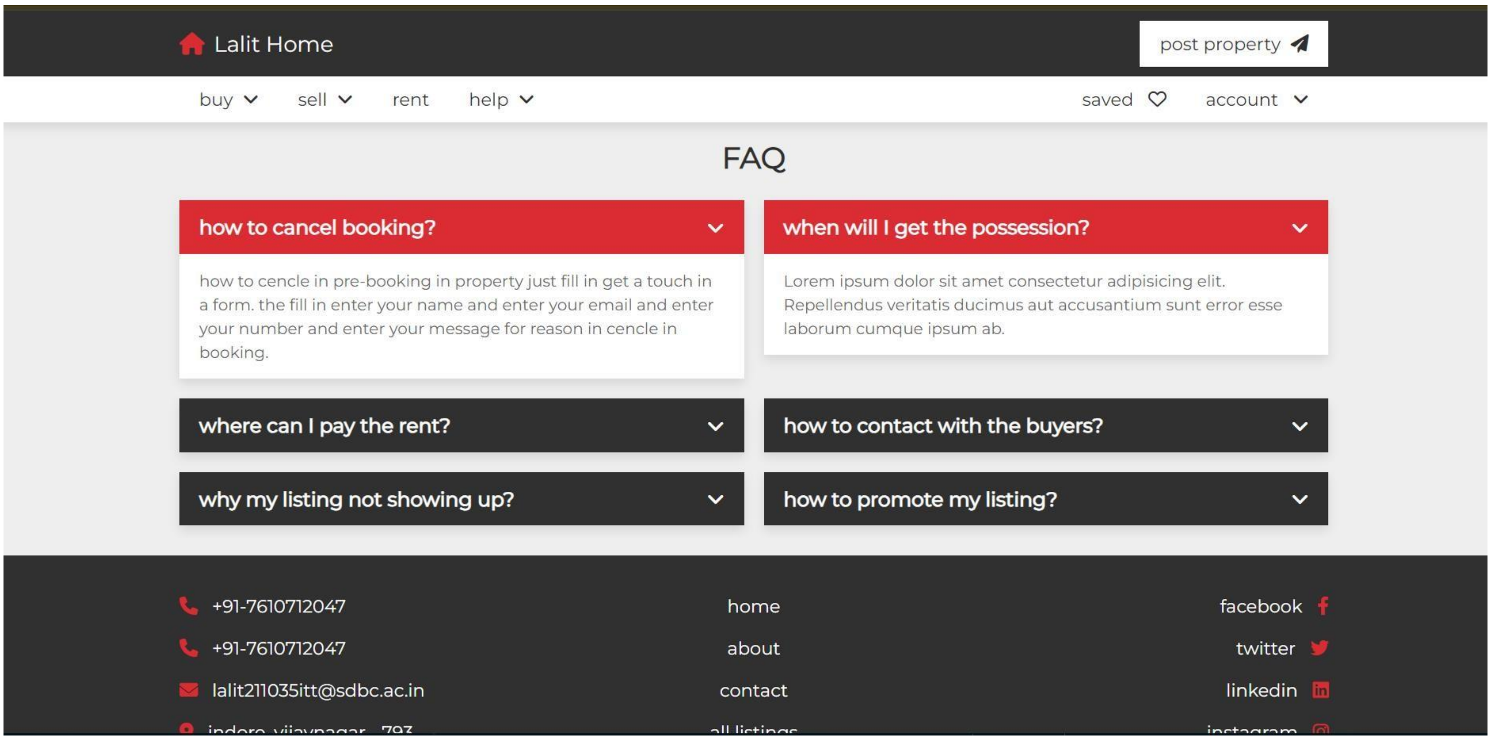
enter your name

enter your email

enter your number

enter your message

Send Message


Lalit Home

post property

buy sell rent help

saved account

FAQ

how to cancel booking?

how to cencle in pre-booking in property just fill in get a touch in a form. the fill in enter your name and enter your email and enter your number and enter your message for reason in cencle in booking.

when will I get the possession?

Lorem ipsum dolor sit amet consectetur adipisicing elit. Repellendus veritatis ducimus aut accusantium sunt error esse laborum cumque ipsum ab.

where can I pay the rent?

how to contact with the buyers?

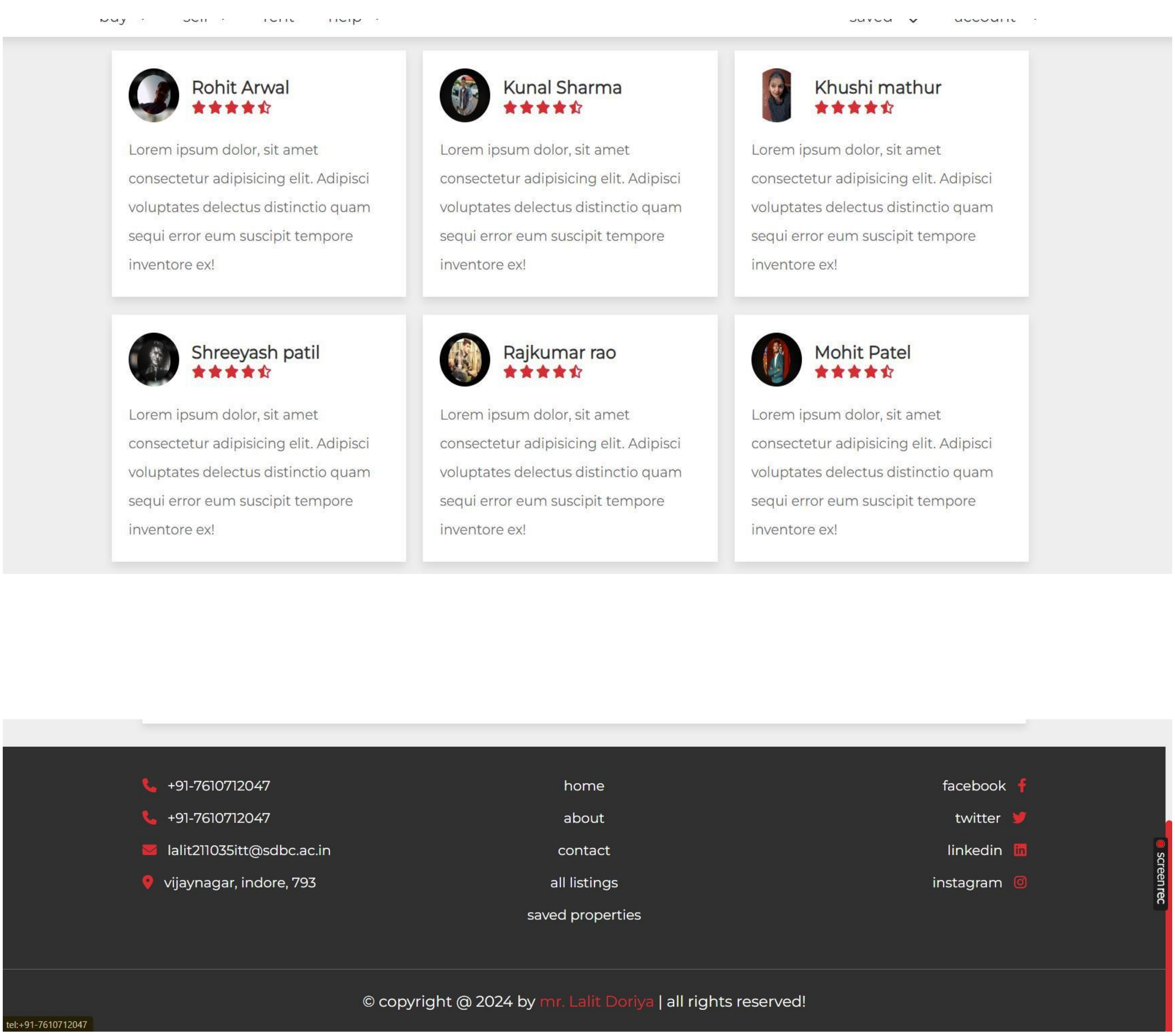
why my listing not showing up?

how to promote my listing?

+91-7610712047
+91-7610712047
lalit211035itt@sdbc.ac.in
indore vijaynagar - 793

home
about
contact
all listings

facebook
twitter
linkedin
instagram



Rohit Arwal
★★★★⯪

Lorem ipsum dolor, sit amet consectetur adipisicing elit. Adipisci voluptates delectus distinctio quam sequi error eum suscipit tempore inventore ex!

Kunal Sharma
★★★★⯪

Lorem ipsum dolor, sit amet consectetur adipisicing elit. Adipisci voluptates delectus distinctio quam sequi error eum suscipit tempore inventore ex!

Khushi mathur
★★★★⯪

Lorem ipsum dolor, sit amet consectetur adipisicing elit. Adipisci voluptates delectus distinctio quam sequi error eum suscipit tempore inventore ex!

Shreeyash patil
★★★★⯪

Lorem ipsum dolor, sit amet consectetur adipisicing elit. Adipisci voluptates delectus distinctio quam sequi error eum suscipit tempore inventore ex!

Rajkumar rao
★★★★⯪

Lorem ipsum dolor, sit amet consectetur adipisicing elit. Adipisci voluptates delectus distinctio quam sequi error eum suscipit tempore inventore ex!

Mohit Patel
★★★★⯪

Lorem ipsum dolor, sit amet consectetur adipisicing elit. Adipisci voluptates delectus distinctio quam sequi error eum suscipit tempore inventore ex!

+91-7610712047
+91-7610712047
lalit211035itt@sdbc.ac.in
vijaynagar, indore, 793

home
about
contact
all listings
saved properties

facebook
twitter
linkedin
instagram

© copyright @ 2024 by mr. Lalit Doriya | all rights reserved!

tel:+91-7610712047

REFERENCES

We have taken references from many resources like YouTube and many websites.

Websites:

- https://www.w3schools.com
- https://www.javatpoint.com
- https://www.codecademy.com
- https://www.stackoverflow.com

YouTube video links:

1. https://youtu.be/5vzCjvUwMXg
2. https://youtu.be/dwVj_g3TpZ4
3. https://youtu.be/L5RpqspNAuc