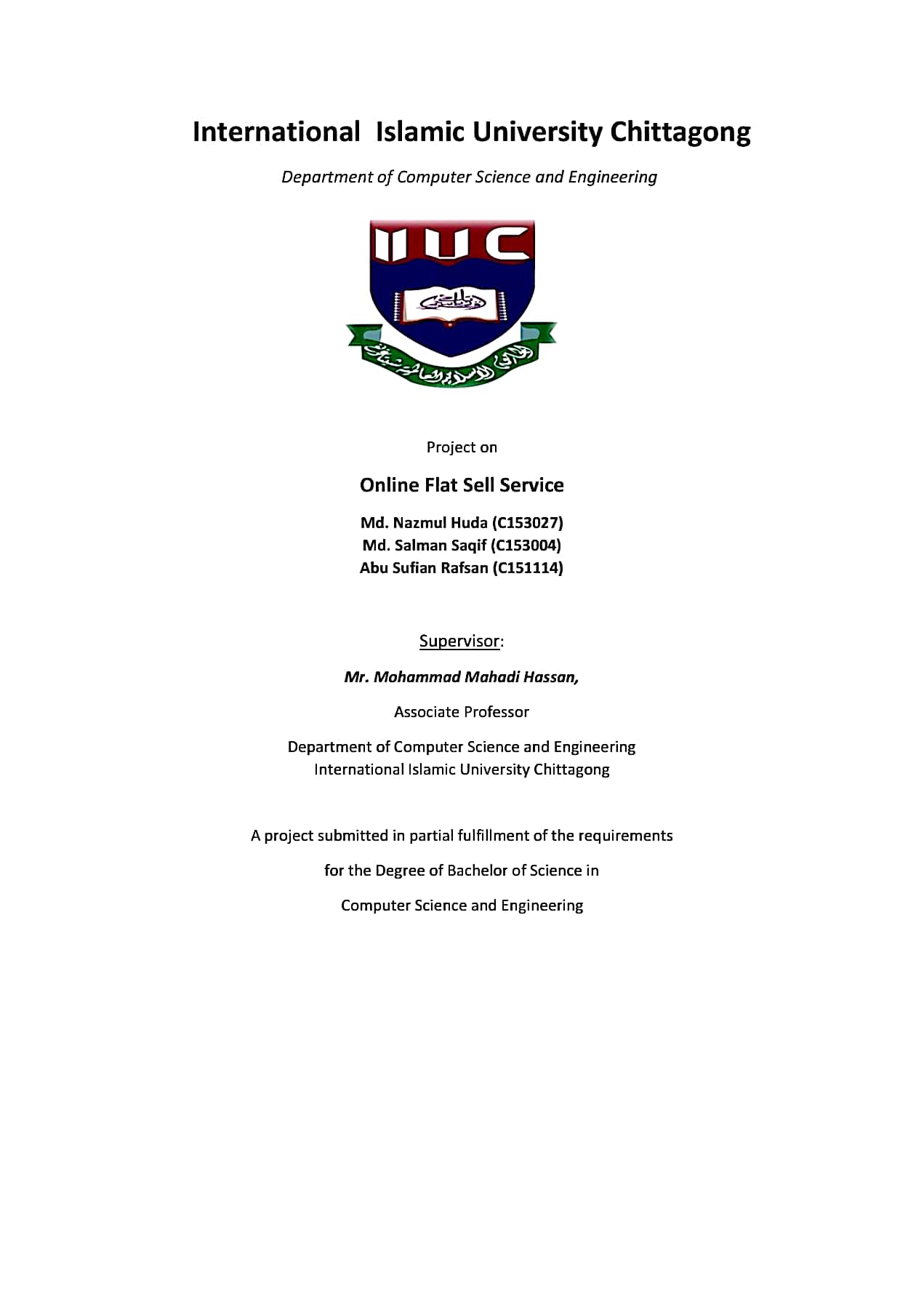
****

**Online Flat Sell Services**

A Report Submitted by

**MD. NAZMUL HUDA ( C153027)**

**MD. SALMAN SAQIF (C153004)**

**ABU SUFIAN RAFSAN (C151114)**

For the partial fulfillment of the degree of

B.Sc in Computer Science & Engineering

Examination held in July 2020

Approved by:

****

**Mr. Mohammad Mahadi Hassan** Supervisor

Associate professor

Department of Computer Science and Engineering

International Islamic University Chittagong, Bangladesh

i

**DEDICATION**

We would like to dedicate our project to our beloved parents for whom our existence in this position today and also our honorable teachers.

ii

**ACKNOWLEDGEMENT**

For every work, motivation, support and proper guidance are most important. It is never possible to do a project individually in perfection. Here we would like to express our deepest gratitude and utmost respect to those who have supported us and helped us throughout during this project. Firstly, we would like to thank Almighty Allah (SWT) for being so merciful that we were capable of doing our project successfully. We would like to thanks our Parents and loved ones for being our constant support throughout. We feel very lucky and honored to show our profound gratitude towards honorable teacher and Supervisor *Mr. Mohammad Mahadi Hassan* for believing us and letting us to do the project. Without his endless support, proper guidance, constructive criticism and helpful suggestions, we would never be able to complete this complicated project. He was so strong just as rousing us to create something that would be helpful for all. In spite of having busy schedule he guided us excellently. We are lucky for having him as our teacher and supervisor. We will cherish this enlightening experience throughout our life. Last but not the least; we would like to thank those people around us along with our family and friends who have constantly provided their critical feedback and support whenever needed. We are likewise grateful to the entirety of our particular educators of our University for their recommendation and participation, which straightforwardly or in a roundabout way helped us to remain, zeroed in on the task.

**iii**

**Abstract**

The world is becoming so fast and peoples are so busy for different purpose of life. Even many people have not enough time to buy something which is important for their daily life. For this reason we have developed a software system to the web to provide flat (location base) to the customer our purpose is to develop this system for buyers and sellers to buy and sell flats (location base) and minimizing their time. This software is also the good e-commerce software and businessman and buyer will be benefited from that.

To recapitulate in short, our main goal is to reduce the stress that people got from buying flats. We will build a online flat market place so that the people can buy flat easily by staying at home. This wont put people in jeopardy and people can buy flat without any beadledom. People will also find all flat informations like Interior design, contact with architecture etc in our website.

**iv**

**Table of Contents**

**CHAPTER-1 INTRODUCTION**

1.1 Introduction......…................................................................................................................1

1.2 Motivation...........................................................................................................................1

1.3 Objective.............................................................................................................................1   
  
**CHAPTER-2 LITERATURE REVIEW** 2.1 Overview…………………………………………………………………………………………………………………………2  
 2.2 Existing system review…………………………………………………………………………………………………….2  
 2.3 Tools……………………………………………………………………………………………………………………………….2

**CHAPTER-3 METHODOLOGY  
3**.1 Necessity of methodology………………………………………………………………………………………………3

3.2 Software development life cycle……………………………………………………………………………………..3

3.3 Results…………………………………………………………………………………………………………………………….3  **CHAPTER-4 SOFTWARE PROCESS**

4.1 Process Model.....................................................................................................................4

4.2 Necessity of Software Process Mode..................................................................................4

4.3 A particular process model for online flat selling...............................................................4

4.4 Result..................................................................................................................................5

**CHAPTER-5 REQUIREMENT ANALYSIS**

5.1 Functional and Non-functional requirement......................................................................6

5.2 User requirement................................................................................................................7

5.3 System requirement...........................................................................................................7

5.4 Result..................................................................................................................................7

**CHAPTER-6 REQUIREMENT ENGINEERING**

6.1 Requirement Engineering Process Model............................................................................8

6.2 Requirement Analysis..........................................................................................................8

6.2.1 Methods For Requirement Analysis...................................................................11

6.2.1 Scenarios............................................................................................................11

6.3 Result.................................................................................................................................11

**CHAPTER-7 SOFTWARE PROCESS**

7.1 Conduct Model..................................................................................................................12

7.2 CASE Workbenches............................................................................................................12

7.3 Data Dictionary..................................................................................................................14

7.4 Result.................................................................................................................................15

**CHAPTER-8 Software Prototyping**

8.1 Prototyping in The Software Process.................................................................................16

8.2 A Prototyping For Online Fat Selling..................................................................................17

8.3 Fast Prototyping Strategy .......................................................…......…...............................17

8.4 Result.................................................................................................................................17

**CHAPTER-9 SOFTWARE SPECIFICATION**

9.1 Software Specification.........................................…...........................................................18

9.2 Formal Software Specification...........................................................................................18

9.3 Formal Specification in The Software Cycle.......................................................................19

9.4 Formal Specification Dialects...................................................…..............................…….…19

9.5 Result.................................................................................................................................19

**CHAPTER-10 OVERALL DESIGN**

10.1 Entity Relationship Diagram............................................................................................20

10.2 Data Flow Diagram(level0)...............................................................................................21

10.3 Data Flow Diagram(level1)...............................................................................................22

10.4 Result................................…............................................................................................22

**CHAPTER-11 IMPLEMENTATION**

11.1 Implementation of Design...............................................................................................24

11.2 Help and Message System...............................................................................................24

11.3 Color Guidelines...............................................................................................................24

10.4 Result.............................................................................…...............................................24

**CHAPTER-12 TESTING**

12.1 The Testing Process.........................................................................................................25

12.2 Classification of Testing..............................….…...............................................................26

12.3 Static Testing....................................................................................................................26

12.4 Dynamic Testing...............................................................................................................26

12.5 White Box Testing............................................................................................................26

12.5.1 Flow Graph.......................................................................................................26

12.5.2 Cyclometric Complexity...................................................................................27

12.5.3 Test Log............................................................................................................28

12.6 Black Box Testing.............................................................................................................28

12.6.1 Equivalence Partitioning..................................................................................28

12.7 Result...............................................................................................................................28

**CHAPTER-13 SYSTEM MAINTENANCE**

13.1 Necessity of Maintenances..............................................................................................29

13.2 Several Types of Maintenance...........................................................................…...........29

13.2.1. Restorative Maintenance................................................................................29

13.2.2 Versatile Maintenance...................................................................................30

13.2.3 Perfective Maintenance...................................................................................30

13.3 Maintenance as Fault Repair...........................................................................................30

13.4 Result...............................................................................................................................30

**CHAPTER-14 DOCUMENTATION**

14.1 Necessity Of Documentation...........................................................................................31

14.2 Textual Documentation...................................................................................................31

14.3 Documentation Providing Manual...................................................................................31

14.3.1 Developer Manual...........................................................................................31

14.3.2 User Manual.....................................................................................................31

14.4 Result...............................................................................................................................32

**CHAPTER-15 CONCLUSION**

15.1 Future Plan.......................................................................................................................33

15.2 Conclusion........................................................................................................................33

15.3 Result...............................................................................................................................33

**REFERENCE**..............................................................................................................................34

**APPENDIX**..........................................................................................................................35-42

**LIST OF FIGURE**

Figure 2.2 Software Development life Cycle................................................................2

Figure 3.3 Throw Away Prototyping Process Model....................................................4

Figure 5.1 Requirement Engineering Process...............................................................8

Figure 5.2 Requirement Analysis Process Model.........................................................9

Figure 5.2.1 VORD Method.........................................................................................11

Figure 6.2 An Analysis and Design Workbench...........................................................13

Figure 7.1 Evolutionary and Throw Away Prototyping...............................................16

Figure 8.3 Formal Specification in the Software cycle................................................19

Figure 9.1 ERD (Entity Relationship Diagram)............................................................20

Figure 9.2 DFD (Data Flow Diagram) : 0 level.............................................................21

Figure 9.3 DFD (Data Flow Diagram) : 1st level...........................................................22

Figure 10.1 Implementation of Design Process...........................................................23

Figure 11.1 The Testing Process..................................................................................25

Figure 12.3 Figure of Maintenance as fault repair......................................................30

**Chapter 1**

**Introduction**

**Introduction**

An online based selling and buying software allows the people to access the data according to their privilege from any place without physically going their. It is become more essential for communication between companies.

**Motivations**

The world is becoming faster. The people always want to become new and updated. But many of them has not enough time to buy product or something. So this field has vast opportunity to be developed. Although there are some individual software is using in the web. But they are not so efficient and user friendly. Also the software is not reach and sometimes it occur error. Therefore we become interested to develop this software, which might overcome the problem of the existing system.

**Objectives**

* To design a website of flat sell service.
* To provide the facility of watching/buying/selling flat with details and proper information.
* To ensuring the Security of transaction and information
* User friendly

1

**Chapter: 2**

**LITERATURE REVIEW**

**2.1 Overview:**

Nowadays flat sell system is not user-friendly. Also, most of their works are done

manually. While we are analyzing we have found some existing flat selling systems.

**2.2 Existing system review:**

**2.2.1 Pbazar dot com**: pbazaar.com started in 2011 with a vision to form a virtual property marketplace wherever native and international customers will seamlessly interact to shop for, sell properties throughout Asian country. Pbazaar’s goal is to form a ascendible platform and business model which can support variant interactions and transactions on a everyday and grow the Asian country property market whereas adding larger accountability and potency.

**2.2.2 Bd housing dot com**: BdHousing is an online real estate marketplace in bangladesh. Being the real estate portal in bangladesh, it occupies a platform for property house owners, whether or not individual or company, to push their properties purchasable, rent, & property seekers, who are searching for property purchasable in bangladesh, to search out their homes.

**2.2.3 Bdproperty dot com**: Bproperty is an e-commerce real estate service provider that also

owns UAE property portal Bayutand, Pakistan’s property portal Zameen, Mubawab in Morocco and the newest acquisition of Bayut’s Saudi Arabia.

**2.3 Tools**:

Here is the list of tools that will help developing our system:

**2.3.1 Software Requirements:**

**Programming language:**

* **PHP** is used for creating active and interactive web pages. PHP is extensively used, obtainable and economical various to competitors like Microsoft’s ASP.
* **jQuery** is a JavaScript library designed to clarify HTML DOM (Document 2

Object Model) tree traversal and manipulation, and event handling, CSS

animation, and Ajax. it's accessible, open-source software using the permissive

MIT License.

**Web designing tools:**

* **HTML** hypertext markup language is one kind of markup language for

documents designed which displayed during a web browser.

* **CSS** Cascading Style Sheets Basically, the work of CSS is to describe the HTML elements as well as how they are to be displayed on paper, paper, and mediator in any screen. A lot of work can be saved by CSS. The layout of multiple web pages can be controlled by CSS all at once. CSS file also stores external style sheets.
* **JavaScript** is an object-oriented computer programming language that basically needed to make interactive effects within web browsers.
* **Bootstrap** is an open-source Javascript framework is called bootstrap. It is combined with CSS, HTML and Javascript code. It is designed to help to build components of user interface. It can be called Front-end-framework as well.[7]

**Database**:

* **MySQL** is an open-source relational database management system (RDBMS). SQL means Structured Query Language.

**Operating system**: Windows 7/8/10, Linux, Mac

**Networking** : Online Hosting

**Software packages**: MS-Office, ERDPlus, Draw.io

**Browser**: Mozilla Firefox, Google Chrome.

**2.3.2 Hardware Requirements**:

* **CPU**: Intel Core i3
* **Processor Speed**: 1.99 GHz
* **Total RAM**: 8GB

**3**

**Chapter 3**

**Methodology**

**3.1 Necessity of Procedure**

Programming is a mind boggling antique made by person. The whole perplexing being is created in a bit by bit system, which is called philosophy for that antique. Consequently technique is required for programming so as to construct it with consistency

**3.2 Software Development Life Cycle**

Programming improvement life cycle gives us a review and rules to create quality programming. For “Online flat sell”, I follow Software Development Life Cycle to make it reliable for the users. The figure has been shown

**System Request**

**Feasibility study**

**Maintenance**

**Requirement definition**

**System Specification**

**Testing**

**System Design**

**Programming & Coding**

**Fig 2.2: Software Development Life Cycle**

4

**3.3 Results**

Programmers are followed by the method and that is important instruction. Technique has been simplified and implementing method has been invented. Now development of the software can go through the above process.

**Chapter 4**

**Software Process**

**4.1 Process Model**

Process is an activity, which operates on an object and changes its state .Model is the graphical representation of an object. So a process model shows the activities of software graphically.

**4.2 Necessity of Process Model**

We need process model because process is important than product (software). If the process is good, the flat also is good

**4.3 A particular Process Model for  *“Online Flat Selling”***

While developing the “Online Flat Selling”, I follow Prototyping Process Model, particularly throw away Prototyping Process Model that is shown in below:

**Design**

**Maintain**

**Test**

**Implement**

**Software Specification**

**Test**

**Design Implement**

**Prototype Specification**

5

**Review prototype**

**Fig 3.3 Throw Away Prototyping Process Model**

I prefer Throw Prototyping Process Model, because of its following advantage:

1. Requirement become clear as customer actually use the system
2. Misunderstanding are clarified.
3. Missing functions are clarified.
4. Poor interface can be refined.
5. Missing requirement can be identified
6. Proper working system is produced, although it is limited in functionality and performance

3) Prototype become specification.

**4.4 Results**

The above system will provide the definite structure to develop the software and proceed. That system will also provide which Technique will be efficient for the software to get most advantage from it.

6

**Chapter 5**

**Requirement Analysis**

**5.1 Functional and non-functional requirements**

**Functional requirements:**

Functional requirement for a system describe the functionality or services that the system is expected to provide. They depend on the type of software, which is being developed, the expected users of the software and the type of system, which is being developed.

Functional requirements for a software system may be expressed in a number of different ways. Here are a number of functional requirements for “Online Flat Sell Selling”:

1. The user shall be able to search either all of the initial set of databases or select a subset from it.
2. The system shall provide appropriate viewers for the user to read documents in the document store.
3. Every order shall be allocated a unique identifier, which the user shall be able to copy to the accounts permanent area.

**Non-functional Requirement :**

Non-functional Requirements are used to define system properties and constraints. These types of requirements arise through user needs because of external factors that may control software environment.

For the “Online Flat Selling”, the non-functional requirements are :

*Flat Requirement:* Flat Requirement incorporates how proficiently and reliabilively, we can utilize our framework, regardless of whether our framework is anything but difficult to compact or not. 7

*Authoritative Requirements:* Authoritative requirement includes how quickly the system delivered services by maintaining standard.

*External Requirement:* External requirement may include security, privacy and safety of the software.

**5.2 User requirements**

The user requirement for "Online Flat Selling " is described by considering both functional and non-functional requirements so that they are understandable by system users who don't

have detailed technical knowledge. User requirements are separated from more detailed system requirements in a requirement document.

**5.3 System requirements**

System requirements are more detailed description of the use requirement. In principle, the system requirement should state what the "system should do and not how it should be Implemented. The system requirements specifications may includes different model Of system such as object model or a data-flow model that is declared later.

**5.4 Results**

After collection of requirement the development of the software become easy and faster. If analysis of the requirement is finished we can go to the secondary Step of the development.

8

**Chapter 6**

**Requirement Engineering**

**6.1 Prerequisite Engineering Process Model**

To create specific programming, everybody ought to follow all the cycle of prerequisite designing. We can characterize prerequisite designing as follows**:**

“Prerequisite Engineering is that movement that changes the necessities and wishes of clients and expected clients of mechanized frameworks, normally deficient and communicated in casual terms in to finish, exact and predictable particulars, ideally written in formal documentations”.

We can diagrammatically speak to necessity building measure as the beneath figure:

User feedback

**User**

**Problem Domain**

**Approval**

**Detail**

**Elicitation**

User Requirement

Requirement specification Models to be valid by user

Knowledge Requirement Model

Request more knowledge Validation Result

Domain knowledge Domain 9

knowledge

**Fig 5.1 Prerequisite Engineering Process**

There are three cycles considered in necessity building. These are:

a) Requirement Analysis / Elicitation

b) Requirement Detail

c) Requirement Approval

**6.2 Prerequisite Analysis**

Prerequisite Analysis is the way toward procuring (evoking) all the pertinent information expected to deliver a necessity model of a difficult area. It is an iterative cycle, which includes area understanding, prerequisite assortment, order, organizing, compromise, prioritization and approval.

We can diagrammatically speak to prerequisite figure:

**Requirement specification**

**Requirement validation**

**Conflict resolution**

**Prioritization**

**Classification**

**Requirement collection**

**Domain understanding**

System entry

10

**Fig 5.2: Prerequisite Analysis**  **Process Model**

Now we can briefly discuss the steps of requirement analysis process model in case of "Online Flat Sell Service " as follows:

**Domain Understanding:**

Domain understanding is concerned with gathering knowledge about the area in which the software system will be applied. The domain for the "Online Flat Selling " concerned with customers, administrator, vendors, flats and maintaining database for these.

**Requirement collection:**

Requirement collection is concerned with the collection of requirement from the user, customer and from knowledge Of problem domain.

**Classification:**

The activity takes the unstructured collection of requirements and organizes them into Coherent clusters is called classification. Two types Of services are given in the Online flat sell service. In one way the buyer may see the flat in the shopping System with details. In another way the buyer may buy flats by entering right username and password.(i.e user account ). 11

**Preoccupation :**

On the off chance that we think about a lot of necessities, there some prerequisite will be a higher priority than others. Prioritization includes association with supporter of find the most significant necessity among the others. In the "Online Flat Sell Service".The administrator may get higher need over the ordinary client to have the level.

**6.2.1 Methods for Requirement Analysis**

For requirement analysis, i uses VORD Method. This method is concerned with Viewpoints. The principal stages of view-point Oriented Requirement Definition (VORD) method are shown in below figure:

**View- point**

**System mapping**

**View- point**

**Documentation**

**View- point**

**Structuring**

**View- point**

**Identification**

Discovering Grouping them Refinement of the

View-point into a hierarchy description of the view points and Services

**Fig 5.2.1 VORD method**

12

**6.2.2 Situations**

Situation can be especially valuable adding point of interest to a layout prerequisite depiction.

At its generally broad, a situation may include:

1) A framework state depiction toward the start of the situation.

2) A depiction of stream of occasions in the situation.

3) A portrayal of what can turn out badly and how this is dealt with

4) Information about different exercises, which may be going on a similar time.

5) A depiction of the condition of the framework after finish of the situation.

**6.3 Results**

In this chapter we have understand requirement analysis, view point identification and even scenarios which is important for the behaviour of the software.

**Chapter 7**

**System Models**

**7.1 Conduct Model**

Conduct models are utilized to depict the general conduct of the framework. This model is two types, named Data flow models & state machine model. These models may be used separately or together, depending on the type of system which is being developed most business systems are primarily driven by data. They are controlled by the data inputs to the system with relatively little external event processing. A data flow model may be all that is needed to represent the behavior of this system.

**7.2 CASE workbenches**

A CASE workbench is a Set of instruments that underpins a specific period of the product

cycle, for example, plan, usage or testing. The benefit of collection CASE apparatuses into a workbench is that devices can cooperate to offer more complete help than is conceivable with a solitary instrument. 13

The following figure shows the devices, which might be remembered for an analysis

and design workbench:

**Central Information Repository**

**Design, analysis and checking tools**

**Query Language Facilities**

**Report generation facilities**

**Structure Diagramming tool**

**Forms Creation Tools**

**Code Generator**

**Data Dictionary**

14

**Fig 6.2 An analysis and design workbench**

**7.3 Data Dictionary**

A data dictionary maintain data about the elements utilized in a framework plan.

Data name : Customer Information

Definition : The product which will served specially, cost will be added.

Data Name : Flat Information

Definition : The product which will served. Specially, cost will be added.

Data Name : Vendors Information

Definition : for the own purchase of the system

15

User Information Table

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Type | Null | Description |
| User\_info id | Int(11) | No |  |
| User\_id | Varchar (32) | No | PK |
| Address type | Char(20) | Yes |  |
| Address\_type\_name | Varchar(32) | Yes |  |
| Company | Varchar(64) | Yes |  |
| Title | Varchar(32) | Yes |  |
| Phone 1 | Varchar(32) | Yes |  |
| Phone 2 | Varchar(32) | Yes |  |
| Address\_1 | Varchar(64) | No |  |
| Address\_2 | Varchar(64) | Yes |  |
| City | Varchar(32) | No |  |

Flat Information Table

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Type | Null | Description |
| Flat\_id | Int(11) | No | Pk |
| Vendor\_id | Int(11) | No |  |
| Flat\_parent\_id | Int(11) | Yes |  |
| Flat\_sku | Varchar(64) | No |  |
| Flat\_S\_desc | Text | Yes |  |
| Flat\_thumb\_image | Varchar(255) | Yes |  |
| Flat\_full\_image | Varchar(255) | Yes |  |
| Flat publish | Char(10) | Yes |  |
| Flat\_length | Metre(55) | Yes |  |
| Flat\_width | Metre(20) | Yes |  |
| Flat height | Metre(10) | Yes |  |
| Flat\_lwh\_uom | Varchar(255) | Yes |  |
| Flat\_url | Varchar(255) | Yes |  |

16

Vendor Information Table

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Type | NULL | Description |
| Vendor\_id | Int(11) | No | PK |
| Vendor\_Name | Varchar(64) | Yes |  |
| Contact\_name | Varchar(32) | Yes |  |
| Contact\_title | Varchar(32) | Yes |  |
| Contact\_phone\_1 | Varchar(32) | No |  |
| Contact\_phone\_2 | Varchar(32) | Yes |  |
| Contact\_email | Varchar(255) | Yes |  |
| Vendor\_phone | Varchar(32) | Yes |  |
| Vendor\_address\_1 | Varchar(64) | No |  |
| Vendor\_address\_2 | Varchar(64) | Yes |  |
| Vendor\_City | Varchar(32) | No |  |
| Vendor\_state | Varchar(32) | No |  |
| Vendor\_country | Varchar(32) | No |  |

**7.4 Result**

Case work tools are important and also data dictionary are required for the various definition of the database. As the database systems are complex that need to simplify as we have mentioned on above.

**Chapter 8**

**Software Prototyping**

**8.1 Prototyping in the software process**

If the systems are large and complex, it is probably impossible to make assessment before the system is built and put into use. One way of tackling this difficulty is to use an evolutionary approach to system development. Prototyping means giving the user a system, which is incomplete and then modifying and augmenting it as the user requirement become clear.

There are two types of prototyping. These are:

l) Evolutionary prototyping

2) Throw away prototyping 17

The following figure illustrates both of these

**Delivered System**

**Evolution Prototyping**

**Outline Requirement**

**Executable prototyping +System Specification**

**Throw away Prototyping**

**Fig7.I: Evolutionary and Throw-away Prototyping**

The objective of Evolutionary prototyping is to deliver a working system to end-users.

The objective of Throwaway prototyping is to validate or derive the system requirements

**8.2 A Prototype for online flat selling**

For "Online flat sell", Throw - away prototyping is used as I mentioned earlier.

**8.3 Fast prototyping strategy**

Fast prototyping strategies are improvement methods, which underline speed of conveyance as opposed to other framework qualities, for example, execution, viability or dependability. There are three quick advancement methods that are reasonable for creating mechanical quality prototype. These are:

1. Dynamic elevated level language improvement
2. Database programming
3. Component and application gathering

I use Dynamic high-level language development technique a prototype for “Online flat sell”.

18

**8.4 Results**

Prototype is used to go through and choose appropriate system that simplify the complex of implementation. We used to prototype as we mentioned earlier above.

**Chapter 9**

**Software Specification**

**Programming Specification**

here are three degrees of programming detail, which might be created.

These are:

l) User prerequisite

2) System prerequisite

3) A product plan determination

The client necessities are the most conceptual determination and the product plan particular is the most nitty gritty. 19

**9.2 Formal Software Specification**

Formal Software Specification can be communicated in a language whose vocabulary,syntax and semantics curve officially characterized. Particular language can't be founded on characteristic language yet on arithmetic. When all is said in done, formal numerical particular falsehood somewhere close to framework prerequisite and programming plan details. They do exclude execution detail yet should introduce a total numerical model of the framework.

**9.3 Formal Specification in the Software cycle**

The accompanying figure shows the phases of programming particular and it's interface with the plan cycle

**Formal Specification**

**System Requirement Specification**

**User Requirement Definition**

**High level Design**

**Architectural Design**

**System Modeling**

**Fig 9.3: Formal Specification in the Software cycle**

**9.4 Formal Specification dialects**

There are two ways to deal with formal particular that have been utilized to compose definite determinations for non-unimportant programming framework. These are:

* An arithmetical methodology where the framework is depicted regarding tasks and their connections.
* A model based methodology where a model of the framework is assemble utilizing numerical builds, for example, sets and arrangement and frameworks activity are characterized by how they alter the framework state.

**20**

**9.5 Results**

Formal numerical determination lie somewhere close to framework necessity and programming plan particular. They do exclude execution detail yet should introduce a total numerical model of the framework**.**

**Chapter 10**

**Overall design**

**10.1 ERD (Entity Relationship Diagram)**

Amount

**Id**

21

**PAY**

**Id**

**Name**

**Id**

**Name**

**Sells**

**Phonee**

**Customer**

**Email**

**Seller**

Address

**Phonee**

**Sells**

Address

**BUY**

**Flat**

**Location**

Address

**Id**

**Fig 10.1: ERD (Entity Relationship Diagram)**

**10.2 DFD (Data Flow Diagram)**

**External Entity**

**Php shop**

**0:level**

**Customer**

**Entity**

**Process**

**Request 22**

Request granted/

Request Rejected

**Fig 10.2 DFD (Data Flow Diagram) : 0 level**

**10.3 DFD (Data Flow Diagram)**

**Input Registration Process**

**Clear process**

**Adding complete**

**Adding process**

**Administrator**

**User**

Request for user

User information information Request

For clear

Registration

23

User information

Clear the registration form

. Request to clear entry

Registration information

**Fig 10.3 DFD (Data Flow Diagram) : 1st level**

**10.4 Result**

Design is very much important to develop a software. Above entity relationship diagram shows actually how the software will be and how it is related to other specification. Dataflow diagram shows how the data is connected to others definition and how it flows to other entity and process the system.

**Chapter 11**

**Implementation**

**11.1 Implementation of Design**

24

While developing the “online Flat Sell software “ we work on the following design process :

**Evaluate design with end users**

**Produced paper based design prototype**

**Analyze and understand user activities**

**Evaluate design with end users**

**Produced dynamic design prototype**

**Design prototype**

**Evaluate prototype**

**Implement final user interface**

.  **Fig 10.1 Implementation of Design process**

The accompanying "UI Design standard" were thought of while building up the product:

* Client Familiarity
* Stability
* Negligible Surprise
* Recover capacity
* User direction
* User decent variety

**11.2 Help and Message System**

If an error occurs, the system provides proper feedback to the user by sending proper help and message. We use polite; concise and consistent error message. We use those terms and concepts, which are drawn from the experience of the people who will make most use of the system. 25

**11.3 Color Guidelines**

While developing the software, we use the following color guidelines:

* We didn't use too many colors.
* We use color-coding consistency to support use tasks.
* We avoid color pairings
* We use color, which shows the status
* We use lower resolution color display

**11.4 Results**

In this part we implemented the software by writing code. And the software gets the real visualization. The software is becoming ready about to use or lunch.

**Chapter 12**

**Testing 26**

**12.1 The testing process**

Testing is a cycle that is utilized to reveal blunders from programming that were made unintentionally as it was planned and built.

We can diagrammatically show various Stags of a testing cycle in underneath:

**System test**

**Confirmation of framework and generally program structure, utilizes black box and**

**Integration testing**

**Last Operational check**

**Approval testing, does it meet the clients desire, discovery test**

**Validation test**

**Test a module's control structure utilizes white box procedures completed by designer**

**Unit testing**

**Fig 12.1: The testing cycle**

**12.2 Classification of testing 27**

There are fundamentally two sorts of testing. These are :

1. Static testing
2. Dynamic testing

Again there are two kinds of dynamic testing. These are:

1. White box testing
2. Black box testing

**12.3 Static testing**

In static testing, program is not run. Another person inspects the code before it is run. For this software, all the codes are reviewed several times.

**12.4 Dynamic Testing**

In dynamic testing, program is executed and test it with test data. Dynamic testing is

Carried out in two ways such as white box testing and black box testing.

**12.5 White Box Testing**

In this sort of testing all the lines are tasted by executing the program. For "On the web

Level Selling", executing the program all the lines are tried. To perform white box testing, premise set testing is applied.

**12.5.1 Stream diagram 28**

We can draw stream diagram for the level Purchase in beneath:

**5**

**4**

**3**

**2**

**1**

**The distinct path 1:**

**4**

**5**

**1**

**3**

**2**

**The distinct Path 2:**

**5**

**4**

.. .. . .

**3**

**2**

**1**

**12.5.2 Cyclometric intricacy**

Cyclometric intricacy is the quantity of ways utilized in the diagram. It is determined by the accompanying condition:

X(Y) = A-B+2

= 5-5+2

=2

Where,

X(Y) = Cyclometric intricacy of the diagram

A= Number

B= Number of hubs

**29**

**12.5.3 Test Log**

This is a log that records the tests applied and the consequences of each test

For the level issue, the test log might be as per the following:

|  |  |  |  |
| --- | --- | --- | --- |
| Stipulation | Consequences | Real effects | Remark |
| Client name=limon  SKU=H01 | False | False | Pass |
| Client name=limon  SKU=H01 | True | True | Pass |
| Client name=limon  SKU=H01 | False | True | Problem bad return |

**12.6 Black Box Testing**

This test is acted in presence of client. Equivalence Partitioning is the technique for black Box Testing.

**12.6.1 Equivalence Partitioning**

In the “online flat selling”, a member can get flat declared by the admin and the member has to log on to the system by id and password.

**12.7 Result**

We discovered a few mistakes while testing and we take care of the issue and the Software become blunder free. This is important to use the software is an efficient way.

**30**

**Chapter 13**

**Maintenance**

**13.1 Need of programming support**

Despite the fact that product doesn't Ware Out, it needs support for the accompanying reasons:

1. It was conveyed bugs. The bugs might be coding mistake, plan blunder or investigation blunder.
2. The condition in Which the product works may change. For instance: the working System may change or equipment stage may change.
3. The client necessity may change. For instance: the product requires new highlights.

Accordingly, programming needs support. As programming "Online level selling" likewise needs support.

**13.2 Several kinds of support**

There are three sorts of support These are:

1. Restorative Maintenance
2. Versatile Maintenance
3. Perfective Maintenance

**13.2.1 Restorative Maintenance**

At the point when bugs show up in the product, these bugs should be right. The cycle of

eliminating bugs is restorative support.

**13.2.2 Versatile Maintenance**

At the point when nature Of the product changes, the product must adjust with its condition. The way toward adjusting with condition is adjusting upkeep

**31**

**13.2.3 Perfective Maintenance**

At the point when the client prerequisite changes, the product must satisfy its client necessity . To satisfy client necessity is perfective upkeep

For Online level selling, any of the three upkeep will be done if essential.

**13.3 Support as Fault Repair**

In reasonable circumstance, the upkeep movement needs to happen direly to react to Customer issues. In this way, When issue happens, we lean toward "Support as Fault Repaid.

Online “flat selling”. The process is shown bellow:

**Derived repair system**

**Modify source code**

**Analyze source code**

**Change request**

**Fig 12.3: Figure of Maintenance as Fault Repair**

**13.4 Results**

Software needs maintenance because it needs to stay longer. Proper maintenance can provide the actual output of software. we can maintain the according to the above articles as we have written.

**32**

**Chapter 14**

**Conclusion**

**14.1 Future Plan:**

The system is still not applicable properly, because it is a big project to do. We will try our best to complete the work properly to make the software useful as we desired. The software will be the great part of the e-commerce. We hope, this software has bright future.

**14.2 Conclusion**

we can say that the developed software played an important role in the field of e-commerce. Customer can easily order their products through internet. It also helps the customer to minimize their valuable time.

Although we have some limitations in Our software but we still hope that both customer and business will be benefited from the software

**14.3 Results**

We have tried our best to provide the improved software and related information. The Software is easy to use and efficient. We have understand many from that project and we will be able to make another big software.

33

**References**:

1. “Software Engineering”.....Ian Somerville, 6th Edition, Addition Wiesly,

August 11, 200

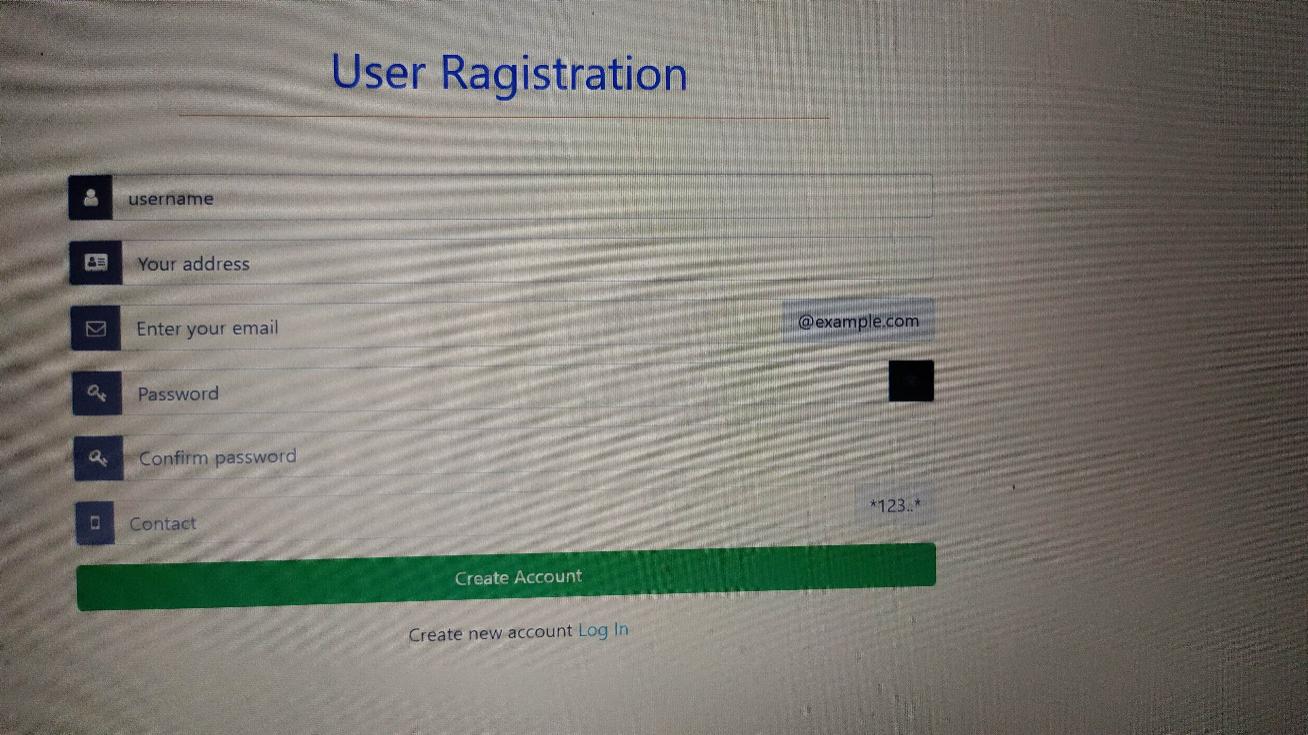
1. “Software Engineering”( A Practitioners Approach) Roger S. Pressman
2. “Database Stream” Pentasoft Publications, United India Colony
3. http://[www.zillow.com](http://www.zillow.com) visited at 19.06.2020
4. <http://www.wikipedia.com>

**34**

**Appendix**

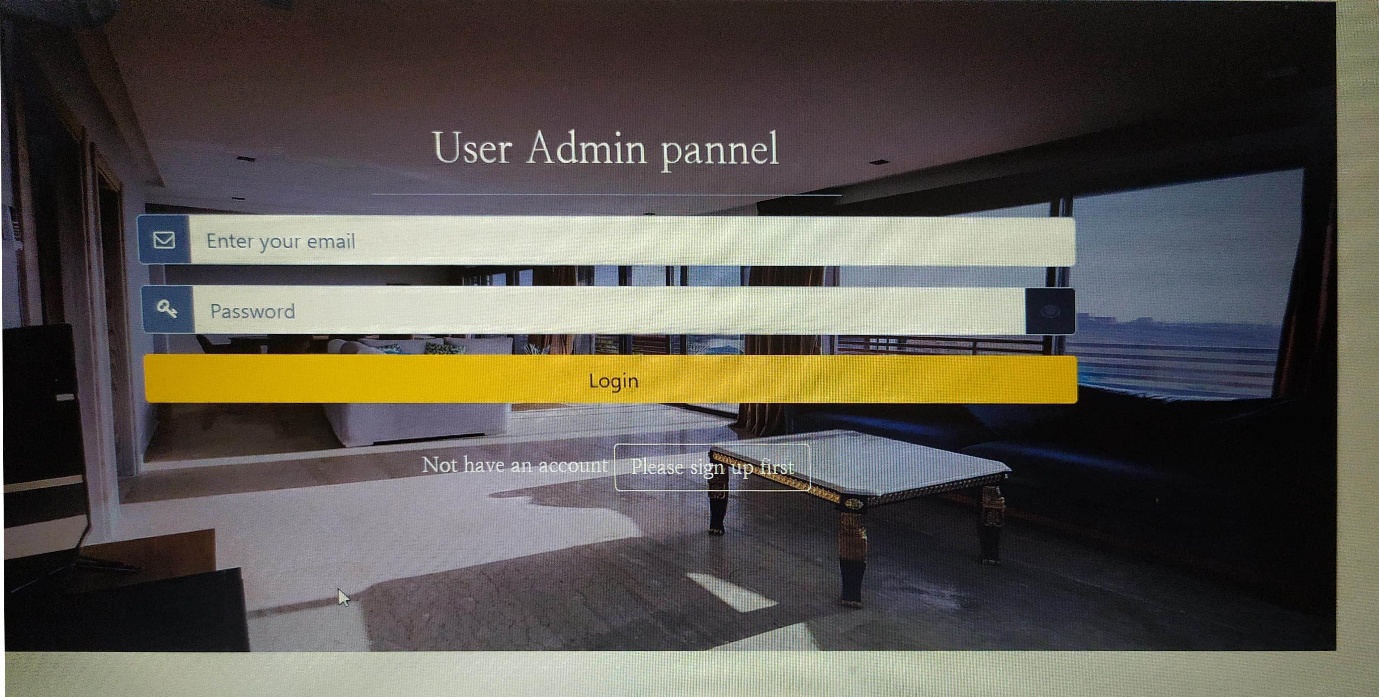
Here we use graphical view of the software to describe how this software work

**User Registration**

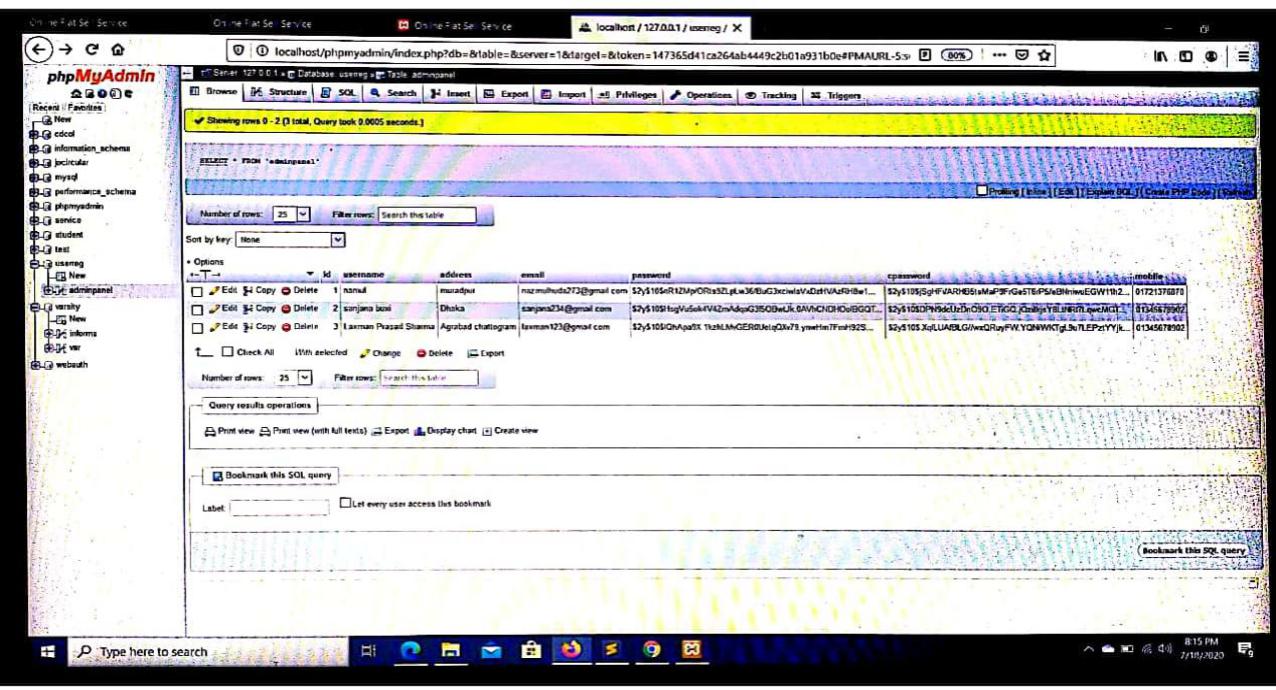


35

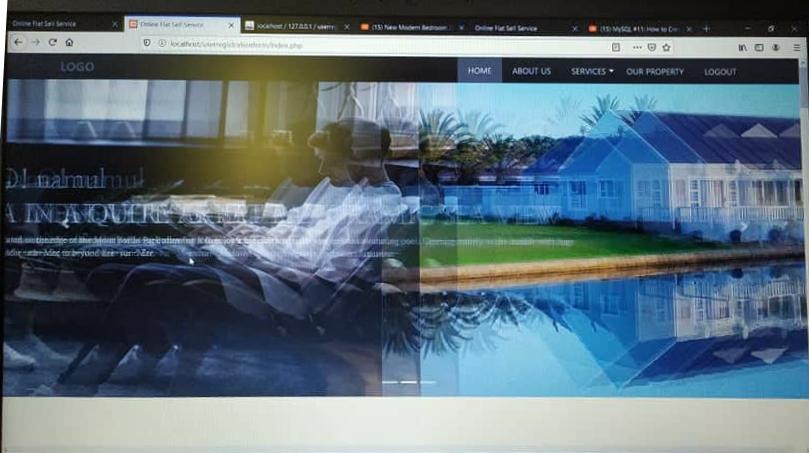
**Admin Panel**

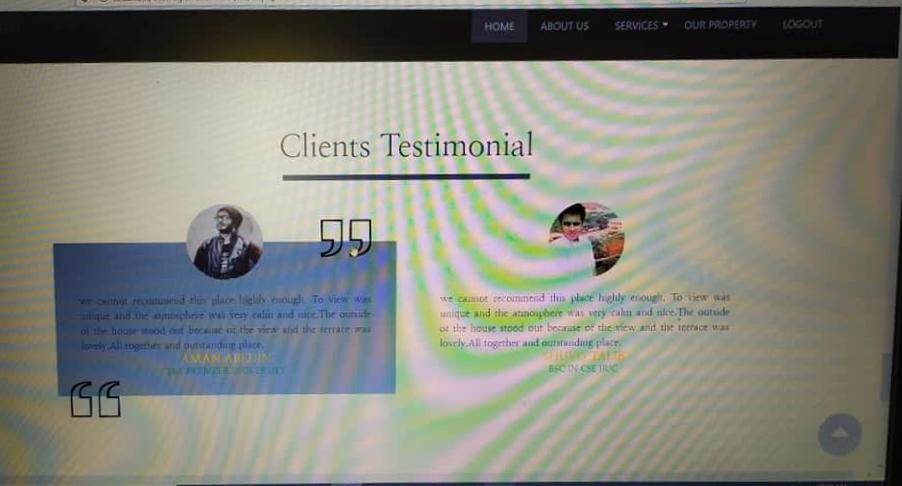
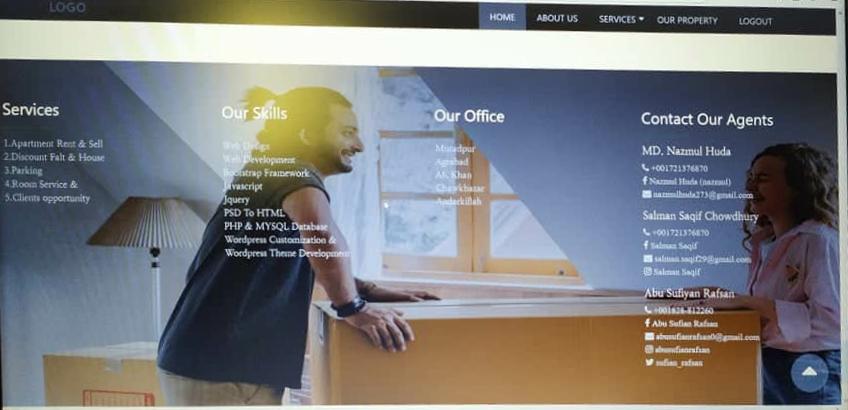


36

Registration Database

37

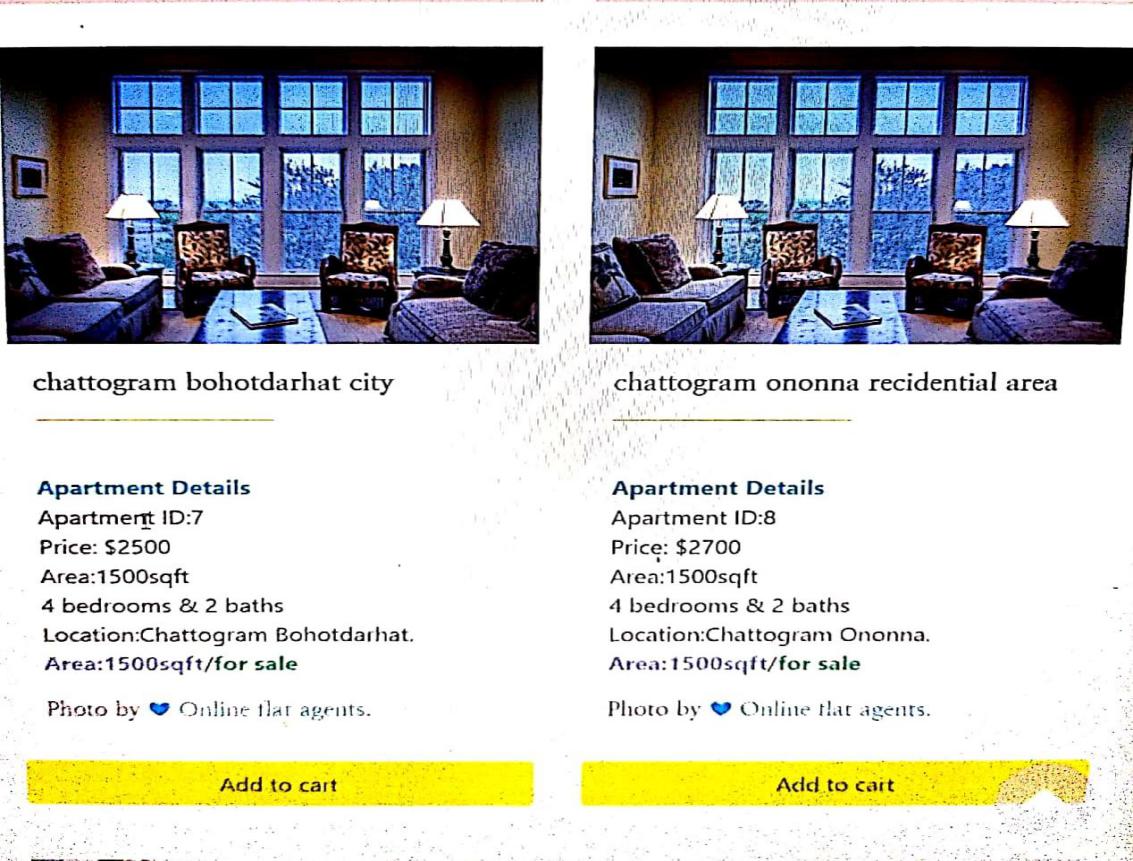
**Home Page**

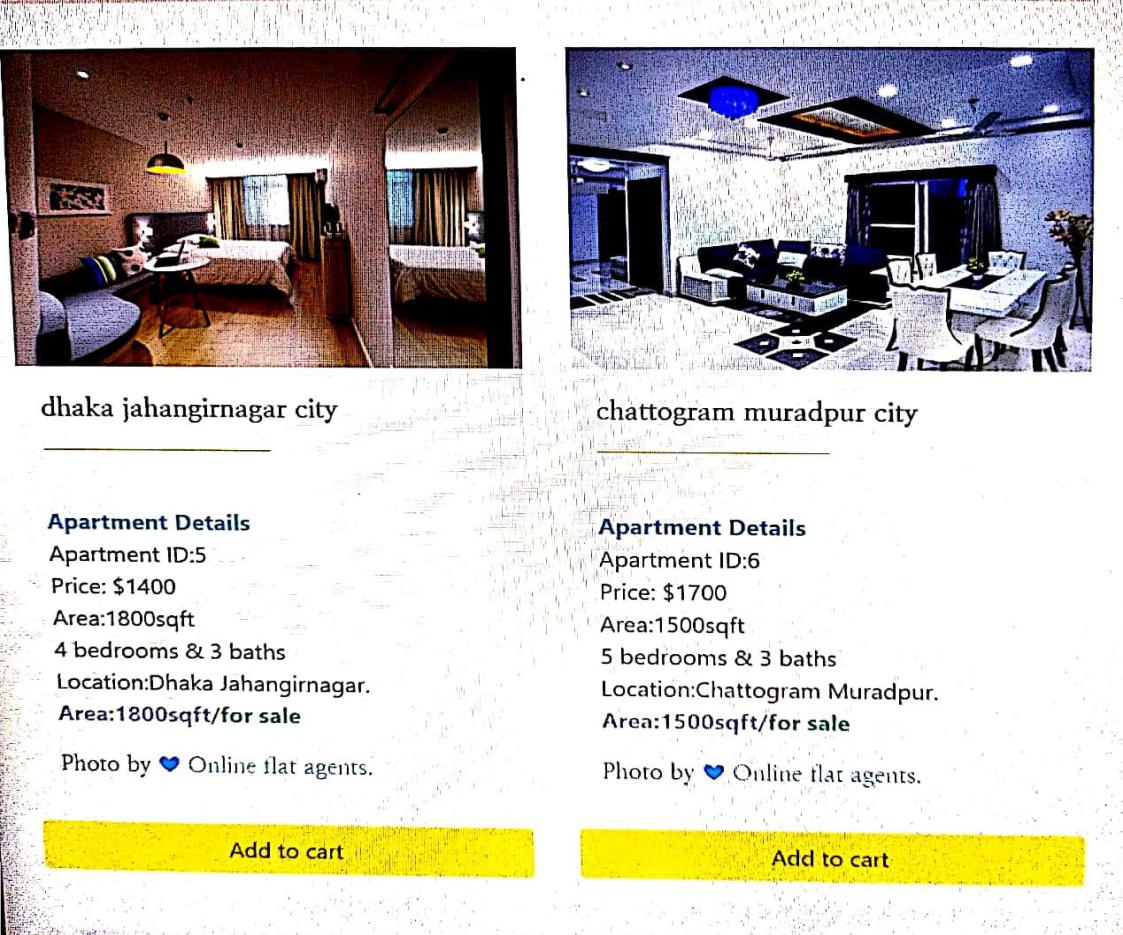
 38



39

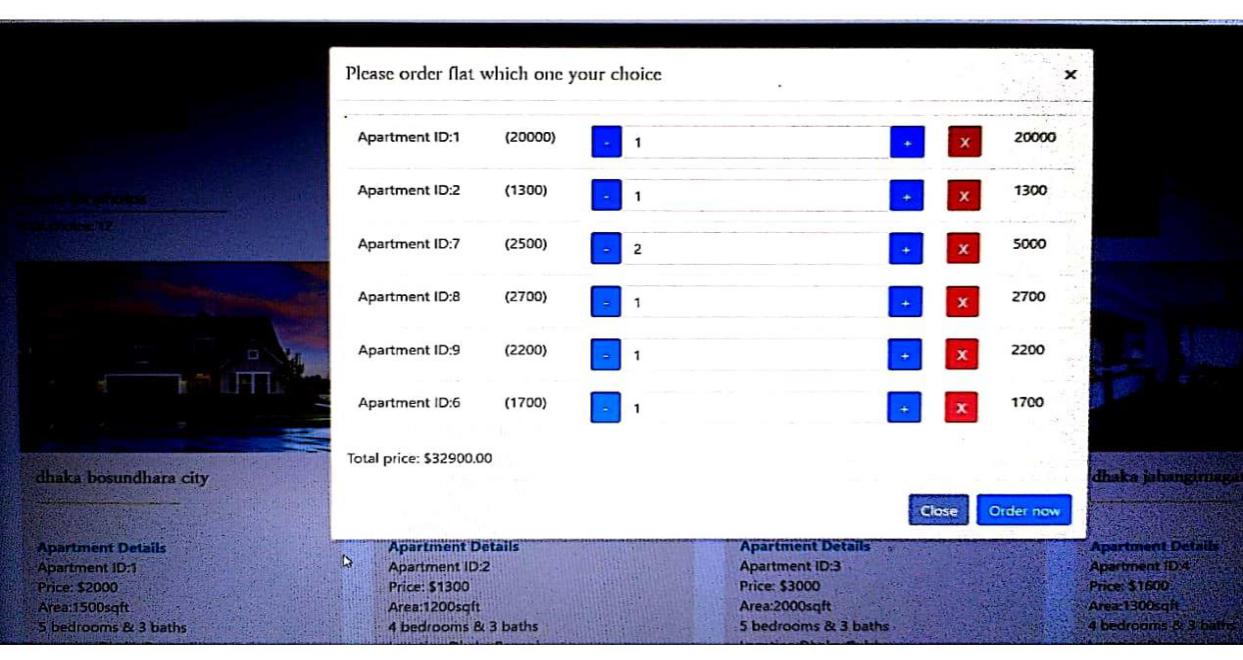
**Flat Details**





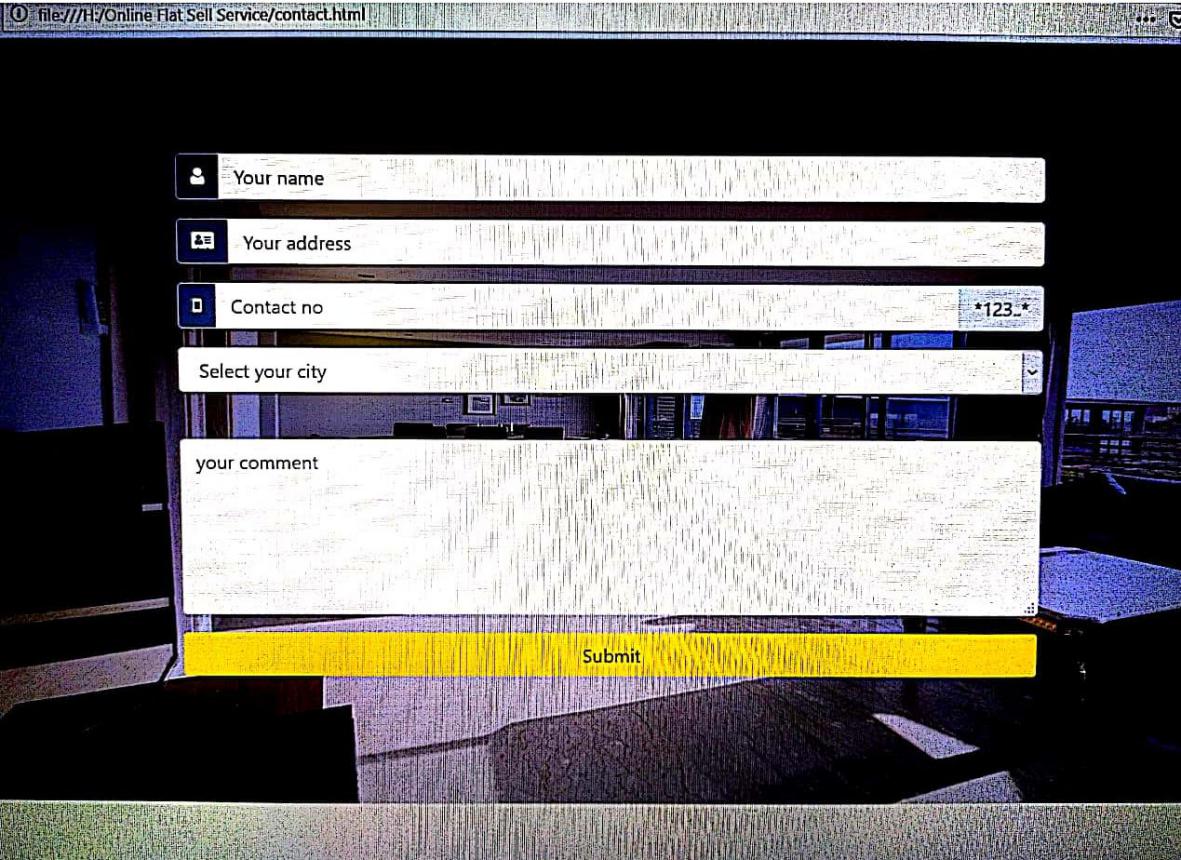
40

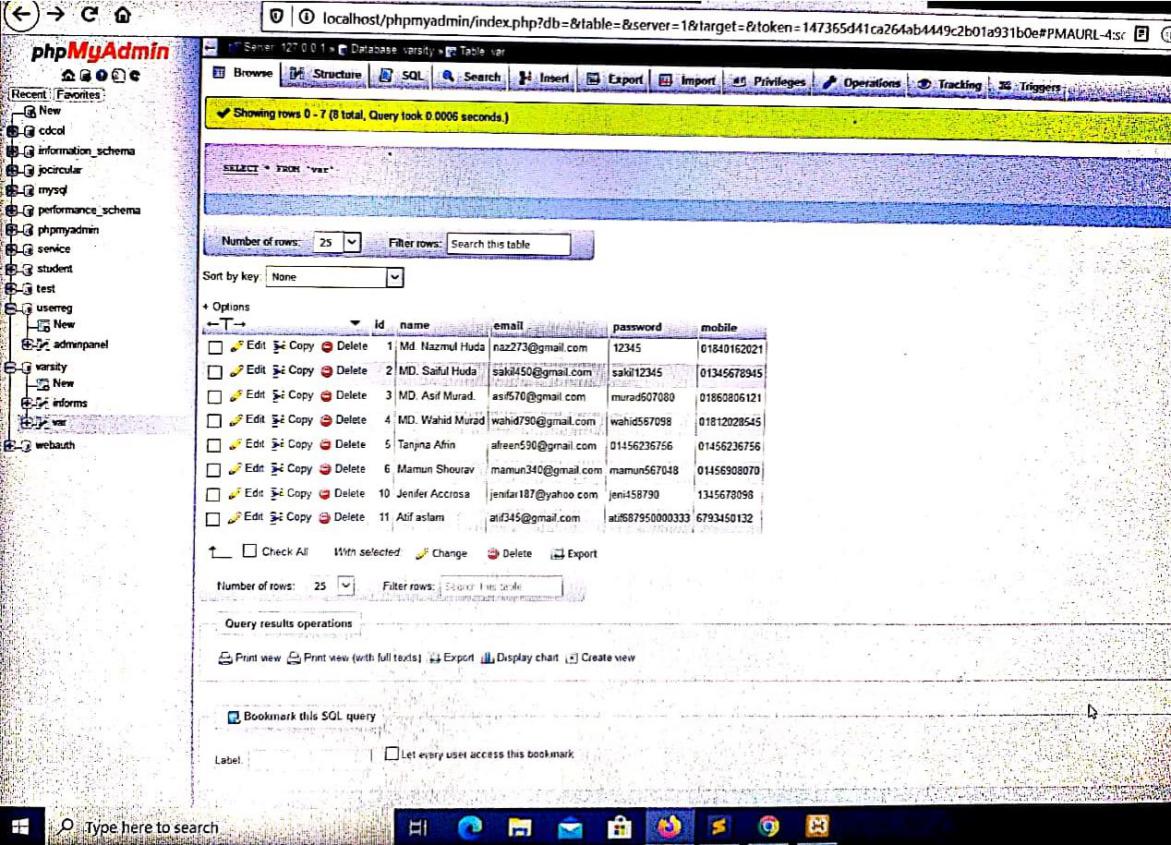
**Click on add to cart for flat order**



41

**Contact us**

****

****

**Database**

**42**