INTRODUCTION

DREAMS is an online website which is designed to help the individuals like house wives, youngsters who are ready to do a home-based business on their own interest. In this website the home-based business owners (seller) can register and use the site as their own website for marketing their products. The customers who wish to buy something which is handmade can search the products or can view the details of individuals (sellers) who are doing business here. The website helps the sellers to improve their marketing and also they can improve the quality and uniqueness of their products from the feedback of the customers

The version control System used during the development time was Git. Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

USING GIT AS A VERSION CONTROL SYSTEM

1. Introduction to GitHub

GitHub is a web-based version-control and collaboration platform for software developers. GitHub, which is delivered through a software-as-a-service (SaaS) business model, was started in 2008 and was founded on Git, an open source code management system created by Linus Torvalds to make software builds faster. And it is used to store the source code for a project and track the complete history of all changes to that code. It allows developers to collaborate on a project more effectively by providing tools for managing possibly conflicting changes from multiple developers. GitHub allows developers to change, adapt and improve software from its public repositories for free, but it charges for private repositories, offering various paid plans. Each public or private repository contains all of a project's files, as well as each file's revision history. Repositories can have multiple collaborators and can be either public or private.

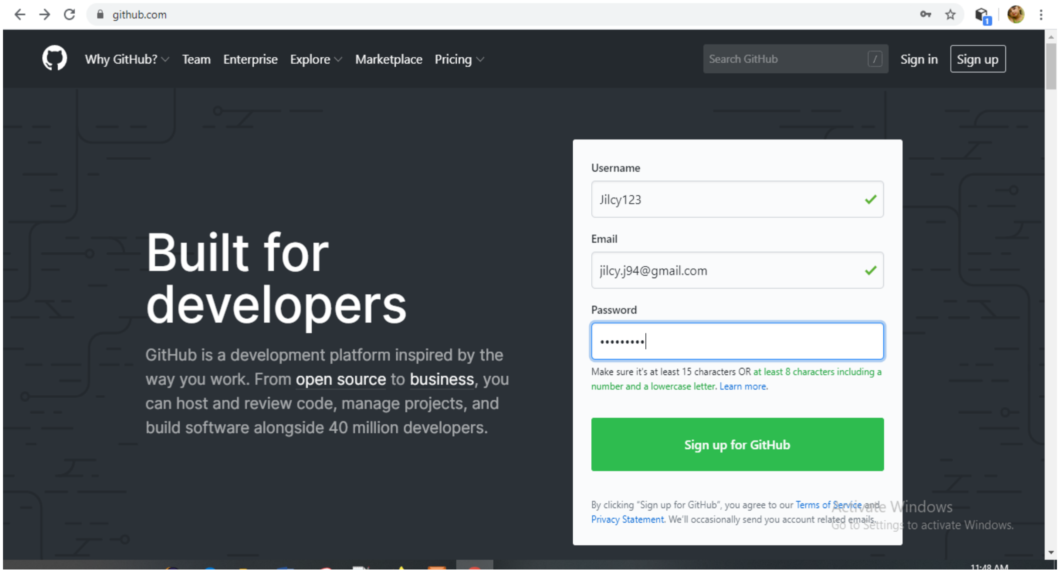
GitHub facilitates social coding by providing a web interface to the Git code repository and management tools for collaboration. GitHub can be thought of as a serious social networking site for software developers. Members can follow each other, rate each other's work, receive updates for specific projects and communicate publicly or privately.

GitHub products and features

GitHub offers an on-premises version in addition to the well-known SaaS product. GitHub Enterprise supports integrated development environments and continuous integration tool integration, as well as a litany of third-party apps and services. It offers increased security and auditable than the SaaS version.

2. Working with Git

- Click on sign up button and create an account in GitHub.com



Built for developers

GitHub is a development platform inspired by the way you work. From open source to business, you can host and review code, manage projects, and build software alongside 40 million developers.

Username: Jilcy123

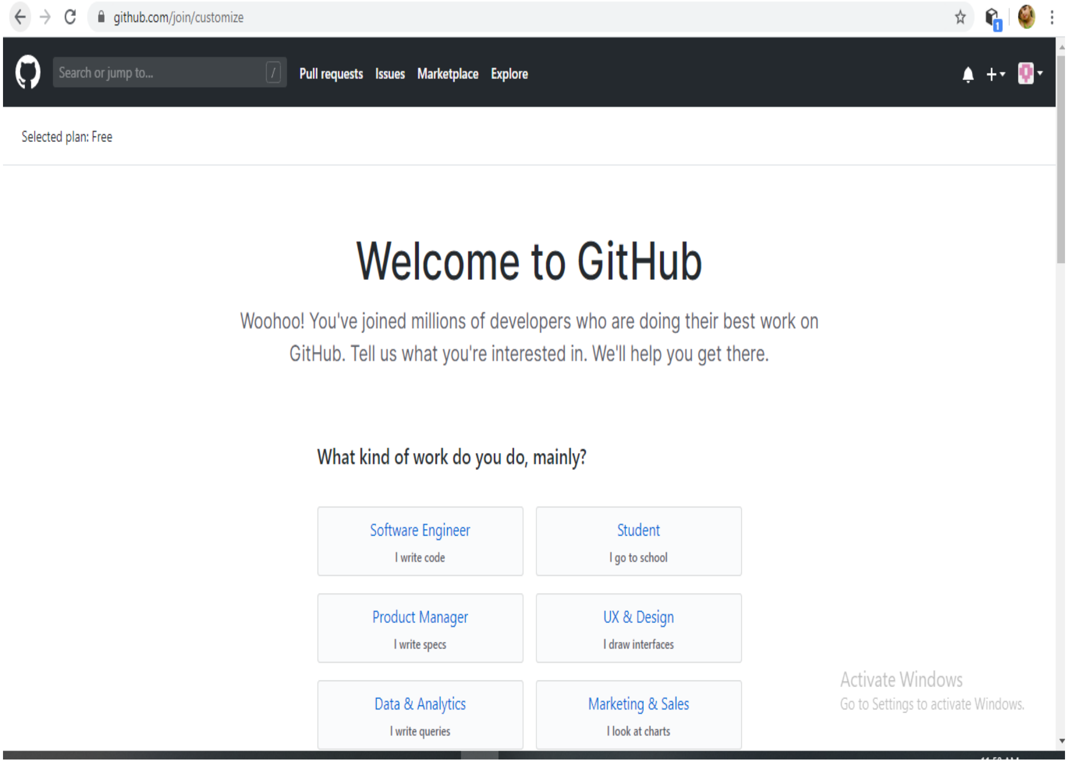
Email: jilcy.j94@gmail.com

Password: •••••••••

Make sure it's at least 15 characters OR at least 8 characters including a number and a lowercase letter. Learn more.

By clicking "Sign up for GitHub", you agree to our Terms of Service and Privacy Statement. We'll occasionally send you account related emails.

- Once successfully sign in, set up personal account, and choose your plan by selecting the options given below based on our purpose.



Selected plan: Free

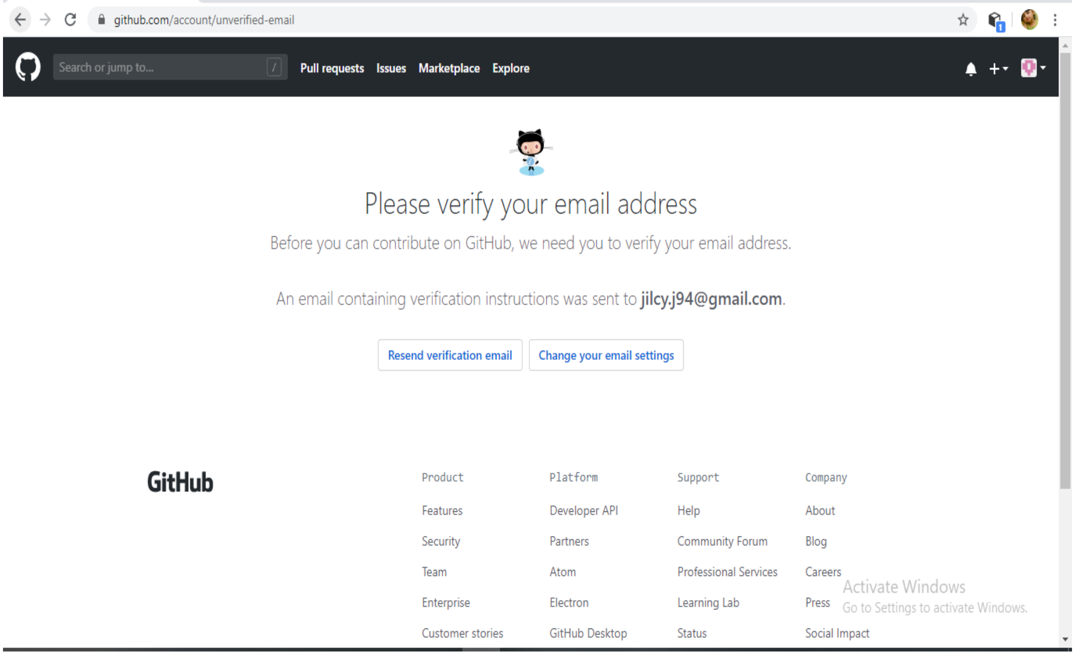
Welcome to GitHub

Woohoo! You've joined millions of developers who are doing their best work on GitHub. Tell us what you're interested in. We'll help you get there.

What kind of work do you do, mainly?

Software Engineer – I write code
Student – I go to school
Product Manager – I write specs
UX & Design – I draw interfaces
Data & Analytics – I write queries
Marketing & Sales – I look at charts

- Once this step is completed then verify the email address and you can access the home page

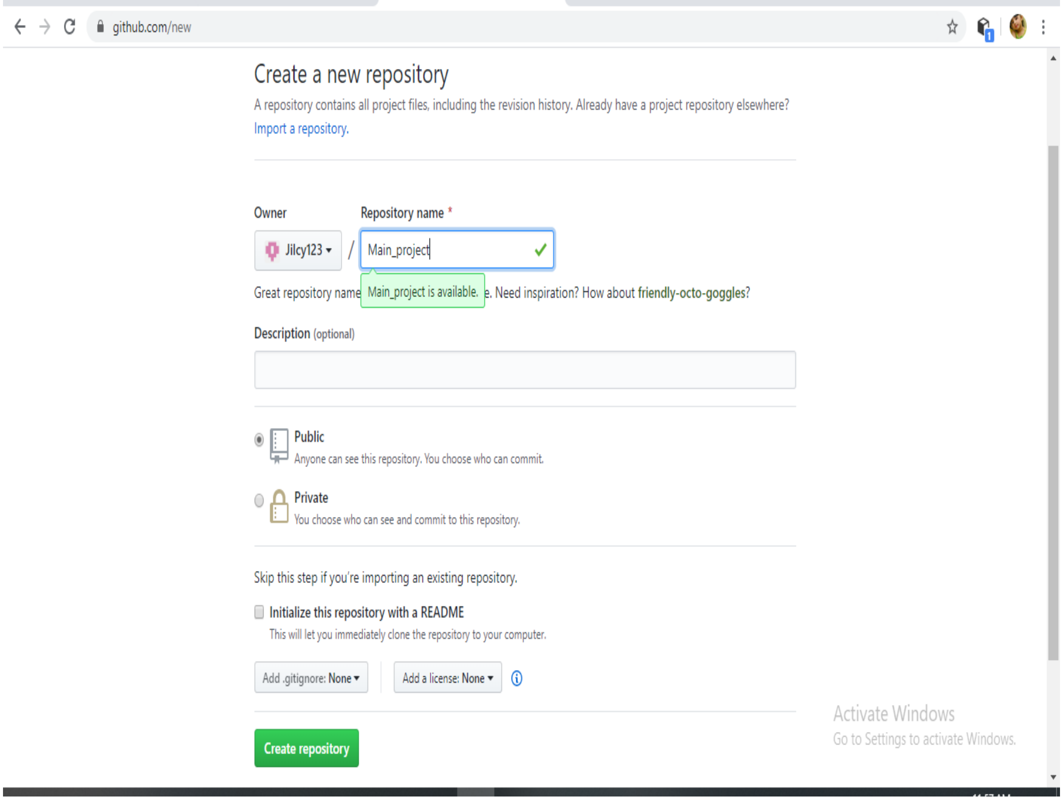


Please verify your email address

Before you can contribute on GitHub, we need you to verify your email address.

An email containing verification instructions was sent to **jilcy.j94@gmail.com**.

- Create a repository in GitHub



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Owner: Jilcy123 / Repository name *: Main_project ✓

Great repository name Main_project is available. e. Need inspiration? How about **friendly-octo-goggles**?

Description (optional)

● Public
Anyone can see this repository. You choose who can commit.

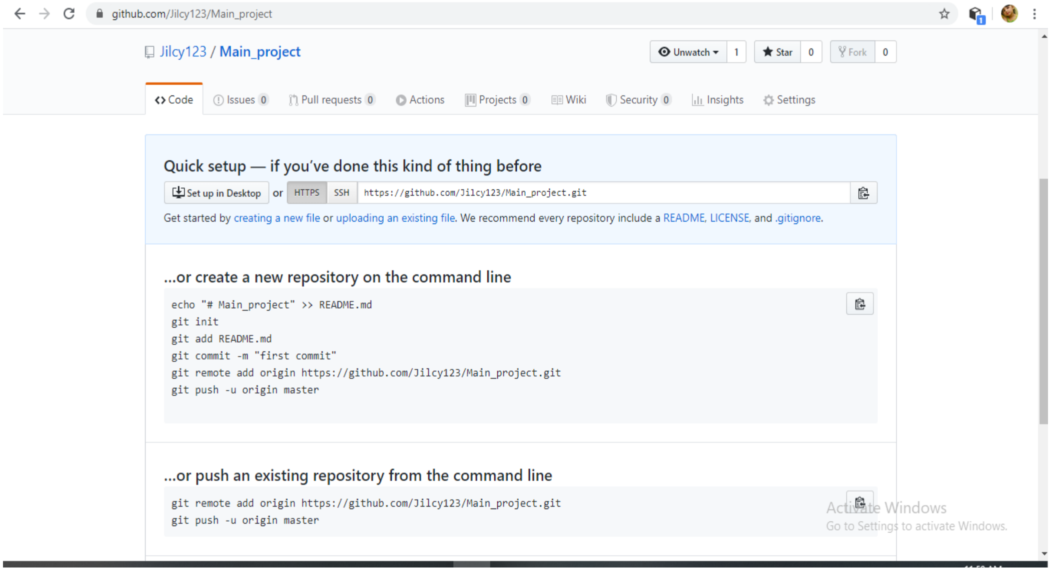
○ Private
You choose who can see and commit to this repository.

Skip this step if you're importing an existing repository.

□ Initialize this repository with a README
This will let you immediately clone the repository to your computer.

Add .gitignore: None ▾ Add a license: None ▾

- Once repository is completed, you can setup the repository



Quick setup — if you've done this kind of thing before

or https://github.com/Jilcy123/Main_project.git

Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE, and .gitignore.

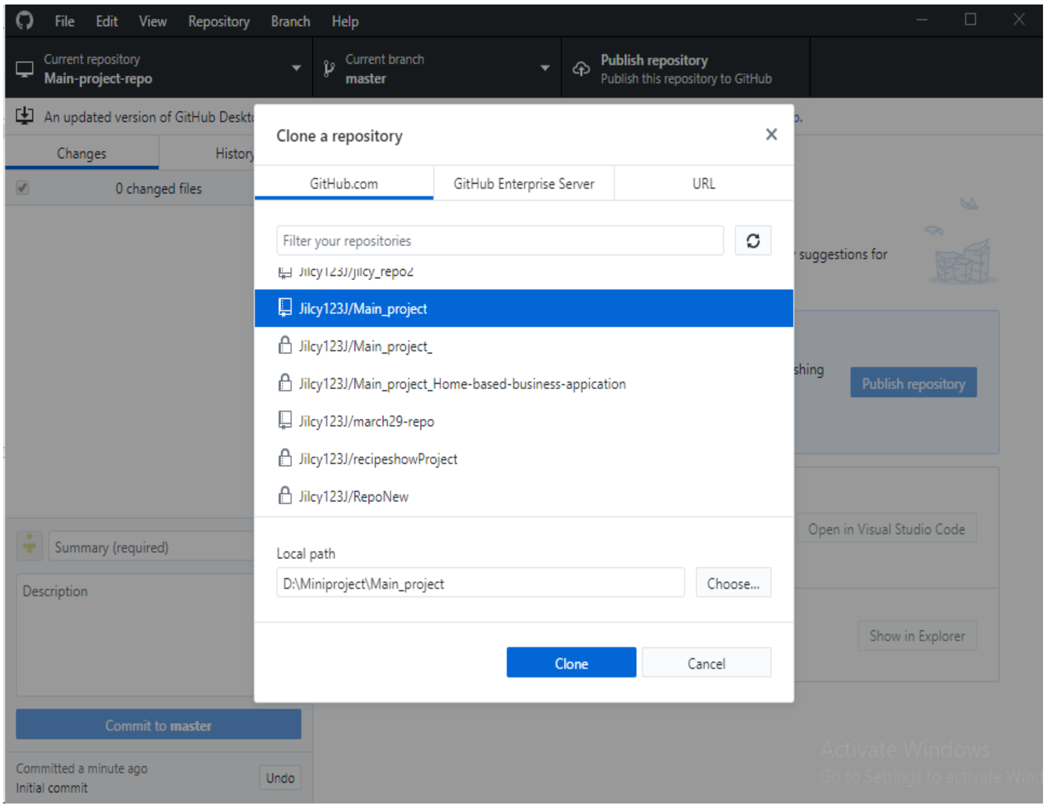
...or create a new repository on the command line

```
echo "# Main_project" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/Jilcy123/Main_project.git
git push -u origin master
```

...or push an existing repository from the command line

```
git remote add origin https://github.com/Jilcy123/Main_project.git
git push -u origin master
```

- Download and install GitHub desktop application. Once installed Go to Repository > Clone repository, and select your repository which was created in GitHub.com or Select a local system folder.



Clone a repository

GitHub.com | GitHub Enterprise Server | URL

Filter your repositories

- Jilcy123/jilcy_repo2
- Jilcy123/Main_project
- Jilcy123/Main_project_
- Jilcy123/Main_project_Home-based-business-appication
- Jilcy123/march29-repo
- Jilcy123/recipeshowProject
- Jilcy123/RepoNew

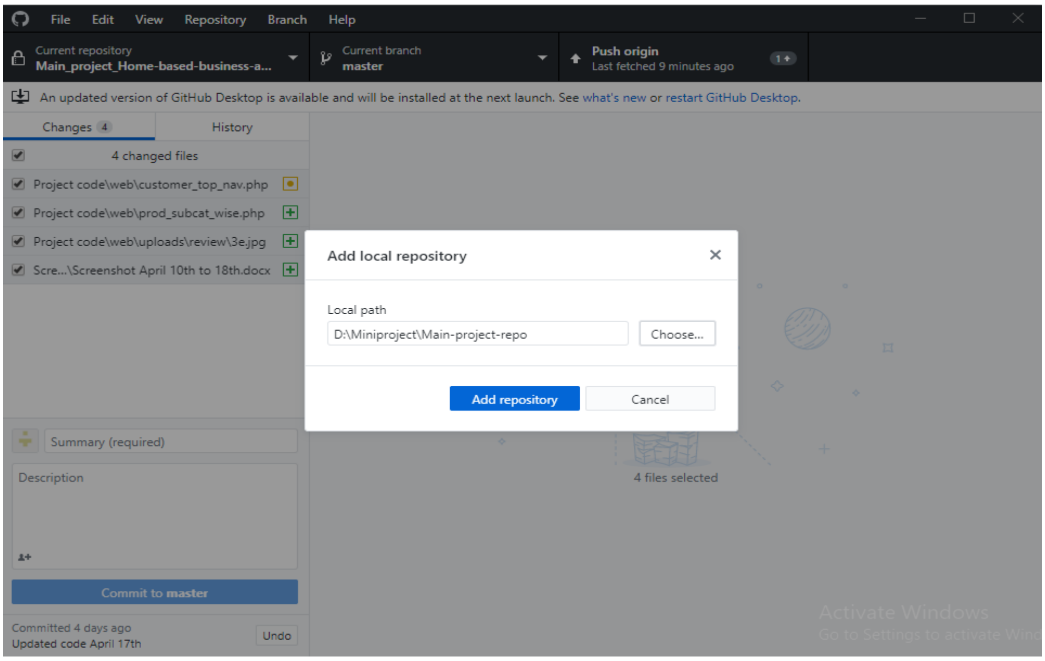
Local path

D:\Miniproject\Main_project

Choose...

Clone | Cancel

□ Add local system folder, then create a local repository with folder name.



Current repository
Main_project_Home-based-business-a...

Current branch
master

Push origin
Last fetched 9 minutes ago

An updated version of GitHub Desktop is available and will be installed at the next launch. See what's new or restart GitHub Desktop.

Changes 4 | History

☑ 4 changed files

☑ Project code\web\customer_top_nav.php

☑ Project code\web\prod_subcat_wise.php

☑ Project code\web\uploads\review\3e.jpg

☑ Scre...\Screenshot April 10th to 18th.docx

Add local repository

Local path

D:\Miniproject\Main-project-repo

Summary (required)

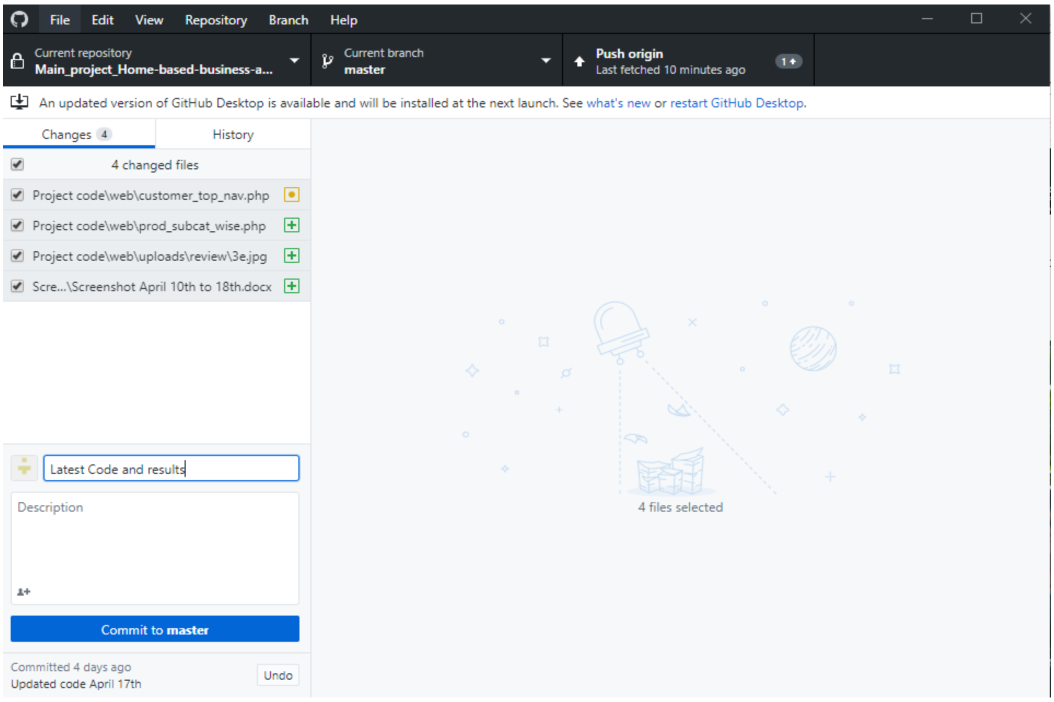
Description

Committed 4 days ago
Updated code April 17th

4 files selected

Activate Windows
Go to Settings to activate Wind

□ Once repository is created, make changes on the file which is stored inside the local repository and commit to master.



Current repository
Main_project_Home-based-business-a...

Current branch
master

Push origin
Last fetched 10 minutes ago

An updated version of GitHub Desktop is available and will be installed at the next launch. See what's new or restart GitHub Desktop.

Changes 4 | History

☑ 4 changed files

☑ Project code\web\customer_top_nav.php

☑ Project code\web\prod_subcat_wise.php

☑ Project code\web\uploads\review\3e.jpg

☑ Scre...\Screenshot April 10th to 18th.docx

Latest Code and results

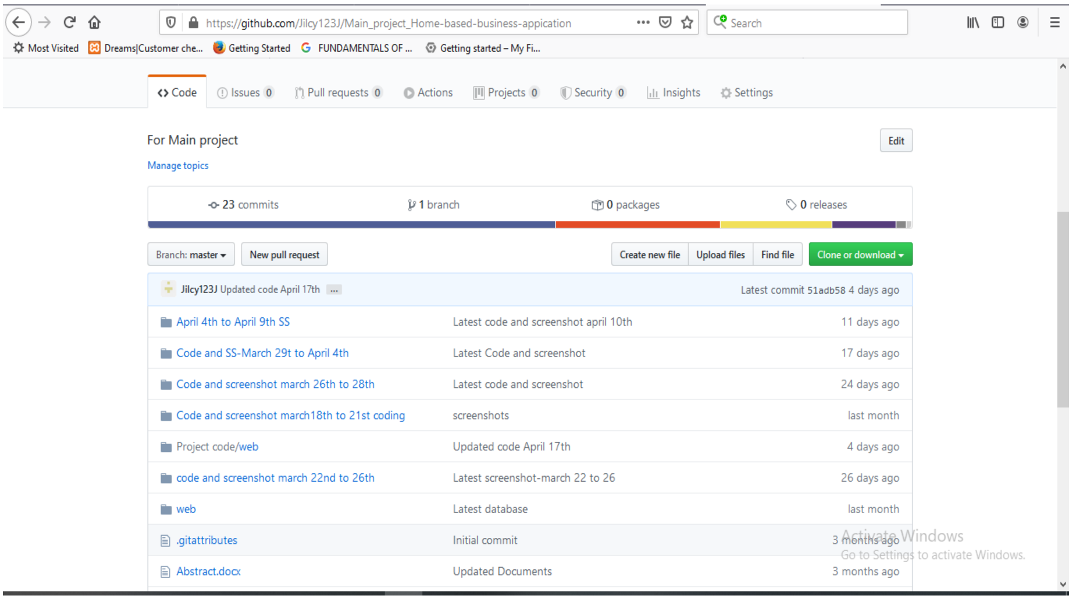
Description

Committed 4 days ago
Updated code April 17th

4 files selected

□ Push the local repository to origin

□ Refresh GitHub.com repository to fetch commits



For Main project

23 commits | 1 branch | 0 packages | 0 releases

Jilcy123J Updated code April 17th ... Latest commit 51adb58 4 days ago

April 4th to April 9th SS	Latest code and screenshot april 10th	11 days ago
Code and SS-March 29t to April 4th	Latest Code and screenshot	17 days ago
Code and screenshot march 26th to 28th	Latest code and screenshot	24 days ago
Code and screenshot march18th to 21st coding	screenshots	last month
Project code/web	Updated code April 17th	4 days ago
code and screenshot march 22nd to 26th	Latest screenshot-march 22 to 26	26 days ago
web	Latest database	last month
.gitattributes	Initial commit	3 months ago
Abstract.docx	Updated Documents	3 months ago

TECHNOLOGY FRAMEWORKS

LARAVEL

Introduction

Laravel is an open-source PHP framework, which is robust and easy to understand. It follows a model-view-controller design pattern. Laravel reuses the existing components of different frameworks which helps in creating a web application. The web application thus designed is more structured and pragmatic.

Laravel offers a rich set of functionalities which incorporates the basic features of PHP frameworks like CodeIgniter, Yii and other programming languages like Ruby on Rails. Laravel has a very rich set of features which will boost the speed of web development.

If you are familiar with Core PHP and Advanced PHP, Laravel will make your task easier. Laravel is accessible, yet powerful, providing powerful tools needed for large, robust applications. A superb inversion of control container, expressive migration system, and tightly integrated unit testing support give you the tools you need to build any application with which you are tasked. It saves a lot time if you are planning to develop a website from scratch. Moreover, a website built in Laravel is secure and prevents several web attacks.

Composer

Composer is a tool which includes all the dependencies and libraries. It allows a user to create a project with respect to the mentioned framework (for example, those used in Laravel installation). Third party libraries can be installed easily with help of composer.
All the dependencies are noted in composer.json file which is placed in the source folder.

Artisan

Command line interface used in Laravel is called Artisan. It includes a set of commands which assists in building a web application. These commands are incorporated from Symphony framework, resulting in add-on features in Laravel 5.1 (latest version of Laravel).

Features of Laravel

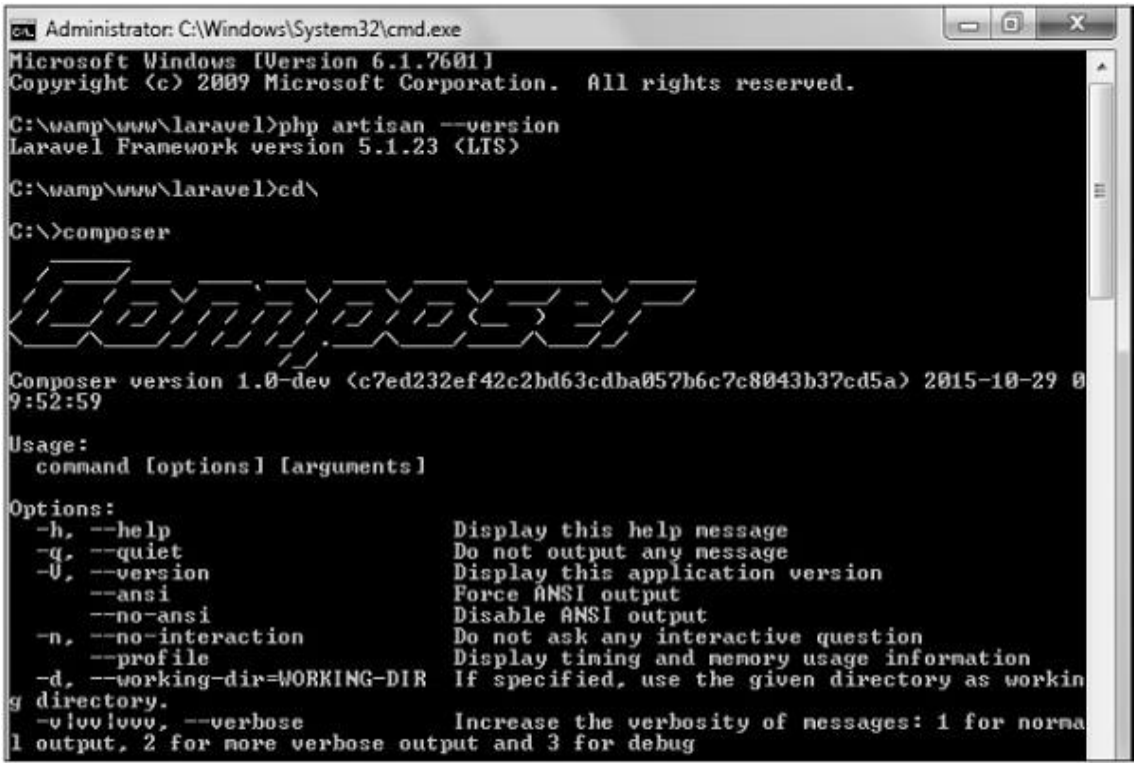
- **Modularity-**Laravel provides 20 built in libraries and modules which helps in enhancement of the application. Every module is integrated with Composer dependency manager which eases updates.

- **Testability-**Laravel includes features and helpers which helps in testing through various test cases**.**
- **Routing-**Routing helps to scale the application in a better way and increases its performance.
- **Configuration Management-**A web application designed in Laravel will be running on different environments, which means that there will be a constant change in its configuration.
- **Query Builder and ORM-**Laravel incorporates a query builder which helps in querying databases using various simple chain methods. It provides ORM (Object Relational Mapper) and ActiveRecord implementation called Eloquent.
- **Schema Builder**
- Schema Builder maintains the database definitions and schema in PHP code.
- **Template Engine**
- Laravel uses the Blade Template engine, a lightweight template language used to design hierarchical blocks and layouts with predefined blocks that include dynamic content.
- **E-mail**-Laravel includes a mail class which helps in sending mail with rich content and attachments from the web application.
- **Authentication**-Laravel eases designing authentication as it includes features such as register, forgot password and send password reminders.
- **Redis**-Laravel uses Redis to connect to an existing session and general-purpose cache.
- **Queues**-Laravel includes queue services like emailing large number of users or a specified Cron jobEvent.
- **Command Bus**-Laravel 5.1 includes Command Bus which helps in executing commands and dispatch events in a simple way. The commands in Laravel act as per the application’s lifecycle.

Implementation of Laravel

- Visit the following URL and download composer to install it on your system using the link https://getcomposer.org/download/

- After the Composer is installed, check the installation by typing the Composer command in the command prompt as shown in the following screenshot.



```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\wamp\www\laravel>php artisan --version
Laravel Framework version 5.1.23 (LTS)

C:\wamp\www\laravel>cd\

C:\>composer
   ______
  / ____/___  ____ ___  ____  ____  ________  _____
 / /   / __ \/ __ `__ \/ __ \/ __ \/ ___/ _ \/ ___/
/ /___/ /_/ / / / / / / /_/ / /_/ (__  )  __/ /
\____/\____/_/ /_/ /_/ .___/\____/____/\___/_/
                    /_/
Composer version 1.0-dev (c7ed232ef42c2bd63cdba057b6c7c8043b37cd5a) 2015-10-29 0
9:52:59

Usage:
  command [options] [arguments]

Options:
  -h, --help                     Display this help message
  -q, --quiet                    Do not output any message
  -V, --version                  Display this application version
      --ansi                     Force ANSI output
      --no-ansi                  Disable ANSI output
  -n, --no-interaction           Do not ask any interactive question
      --profile                  Display timing and memory usage information
  -d, --working-dir=WORKING-DIR  If specified, use the given directory as workin
g directory.
  -v|vv|vvv, --verbose           Increase the verbosity of messages: 1 for norma
l output, 2 for more verbose output and 3 for debug
```

- Create a new directory anywhere in your system for your new Laravel project. After that, move to path where you have created the new directory and type the following command there to install Laravel.

  `composer create-project laravel/laravel your-project-name` 4.2.*

  This command will download and install a fresh copy of Laravel in a new ‘your-project-name’ folder within your current directory.

Permissions

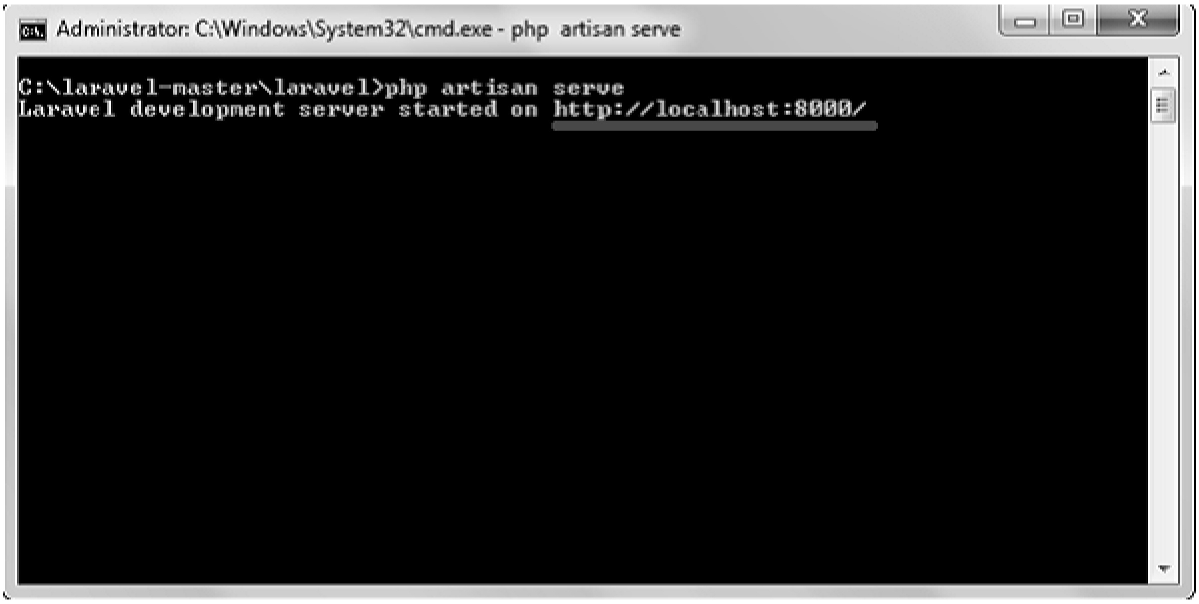
After installing Laravel, you may need to grant the web server write permissions to the app/storage directories.

Serving Laravel

- Start the Laravel service by executing the following command.

  ```
  php artisan serve
  ```

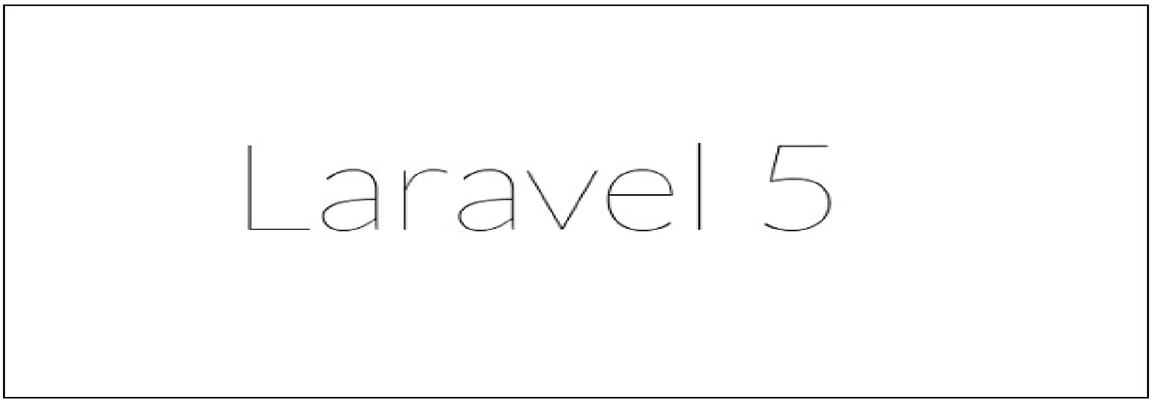
- After executing the above command, you will see a screen as shown below –



Administrator: C:\Windows\System32\cmd.exe - php artisan serve

```
C:\laravel-master\laravel>php artisan serve
Laravel development server started on http://localhost:8000/
```

- Copy the URL underlined in gray in the above screenshot and open that URL in the browser. If you see the following screen, it implies Laravel has been installed successfully.



Directory Structure

After installing the framework, the app directory contains folders such as views, controllers, and models. Most of your application's code will reside somewhere in this directory.

Angular

Introduction

Angular is a platform and framework for building single-page client applications using HTML and TypeScript. Angular is written in TypeScript. It implements core and optional functionality as a set of TypeScript libraries that you import into your apps.

The architecture of an Angular application relies on certain fundamental concepts. The basic building blocks are NgModules, which provide a compilation context for components. NgModules collect related code into functional sets; an Angular app is defined by a set of NgModules. An app always has at least a root module that enables bootstrapping, and typically has many more feature modules.

- Components define views, which are sets of screen elements that Angular can choose among and modify according to your program logic and data.
- Components use services, which provide specific functionality not directly related to views. Service providers can be injected into components as dependencies, making your code modular, reusable, and efficient.

Both components and services are simply classes, with decorators that mark their type and provide metadata that tells Angular how to use them. An app's components typically define many views, arranged hierarchically. Angular provides the Router service to help you define navigation paths among views. The router provides sophisticated in-browser navigational capabilities.

- The metadata for a component class associates it with a template that defines a view. A template combines ordinary HTML with Angular directives and binding markup that allow Angular to modify the HTML before rendering it for display.
- The metadata for a service class provides the information Angular needs to make it available to components through dependency injection (DI).

An app's components typically define many views, arranged hierarchically. Angular provides the Router service to help you define navigation paths among views. The router provides sophisticated in-browser navigational capabilities.

Modules

NgModules differ from and complement JavaScript (ES2015) modules. An NgModule declares a compilation context for a set of components that is dedicated to an application domain, a workflow, or a closely related set of capabilities. An NgModule can associate its components with related code, such as services, to form functional units.

Every Angular app has a root module, conventionally named AppModule, which provides the bootstrap mechanism that launches the application. An app typically contains many functional modules.

Like JavaScript modules, NgModules can import functionality from other NgModules, and allow their own functionality to be exported and used by other NgModules. For example, to use the router service in your app, you import the Router NgModule.

Components

Every Angular application has at least one component, the root component that connects a component hierarchy with the page document object model (DOM). Each component defines a class that contains application data and logic, and is associated with an HTML template that defines a view to be displayed in a target environment.

The @Component() decorator identifies the class immediately below it as a component, and provides the template and related component-specific metadata.

Templates, directives, and data binding

A template combines HTML with Angular markup that can modify HTML elements before they are displayed. Template directives provide program logic, and binding markup connects your application data and the DOM. There are two types of data binding:

- Event binding lets your app respond to user input in the target environment by updating your application data.
- Property binding lets you interpolate values that are computed from your application data into the HTML.

Before a view is displayed, Angular evaluates the directives and resolves the binding syntax in the template to modify the HTML elements and the DOM, according to your program data and logic. Angular supports two-way data binding, meaning that changes in the DOM, such as user choices, are also reflected in your program data.

Services and dependency injection

For data or logic that isn't associated with a specific view, and that you want to share across components, you create a service class. A service class definition is immediately preceded by the @Injectable() decorator. The decorator provides the metadata that allows other providers to be injected as dependencies into your class.

Dependency injection (DI) lets you keep your component classes lean and efficient. They don't fetch data from the server, validate user input, or log directly to the console; they delegate such tasks to services.

Routing

The Angular Router NgModule provides a service that lets you define a navigation path among the different application states and view hierarchies in your app. It is modeled on the familiar browser navigation conventions:

- Enter a URL in the address bar and the browser navigates to a corresponding page.
- Click links on the page and the browser navigates to a new page.
- Click the browser's back and forward buttons and the browser navigates backward and forward through the history of pages you've seen.

The router maps URL-like paths to views instead of pages. When a user performs an action, such as clicking a link that would load a new page in the browser, the router intercepts the browser's behavior, and shows or hides view hierarchies.

If the router determines that the current application state requires particular functionality, and the module that defines it hasn't been loaded, the router can lazy-load the module on demand.

Setting up the local environment and workspace

Prerequisites

To use the Angular framework, you should be familiar with the following:

- JavaScript
- HTML
- CSS

Node.js

Angular requires a current, active LTS, or maintenance LTS version of Node.js. See the engines key for the specific version requirements in our package.json.

- To check your version, run node -v in a terminal/console window.
- To get Node.js, go to nodejs.org.

npm package manager

Angular, the Angular CLI, and Angular apps depend on features and functionality provided by libraries that are available as npm packages. To download and install npm packages, you must have an npm package manager.

To check that you have the npm client installed, run `npm -v` in a terminal/console window.

Implementation

Step 1: Install the Angular CLI

You use the Angular CLI to create projects, generate application and library code, and perform a variety of ongoing development tasks such as testing, bundling, and deployment.

Install the Angular CLI globally.

To install the CLI using npm, open a terminal/console window and enter the following command:

```
npm install -g @angular/cli
```

Step 2: Create a workspace and initial application

To create a new workspace and initial starter app:

1. Run the CLI command `ng new` and provide the name `my-app`, as shown here:

   ```
   ng new my-app
   ```

2. The ng new command prompts you for information about features to include in the initial app. Accept the defaults by pressing the Enter or Return key.

The Angular CLI installs the necessary Angular npm packages and other dependencies. The CLI creates a new workspace and a simple Welcome app, ready to run.

Step 3: Run the application

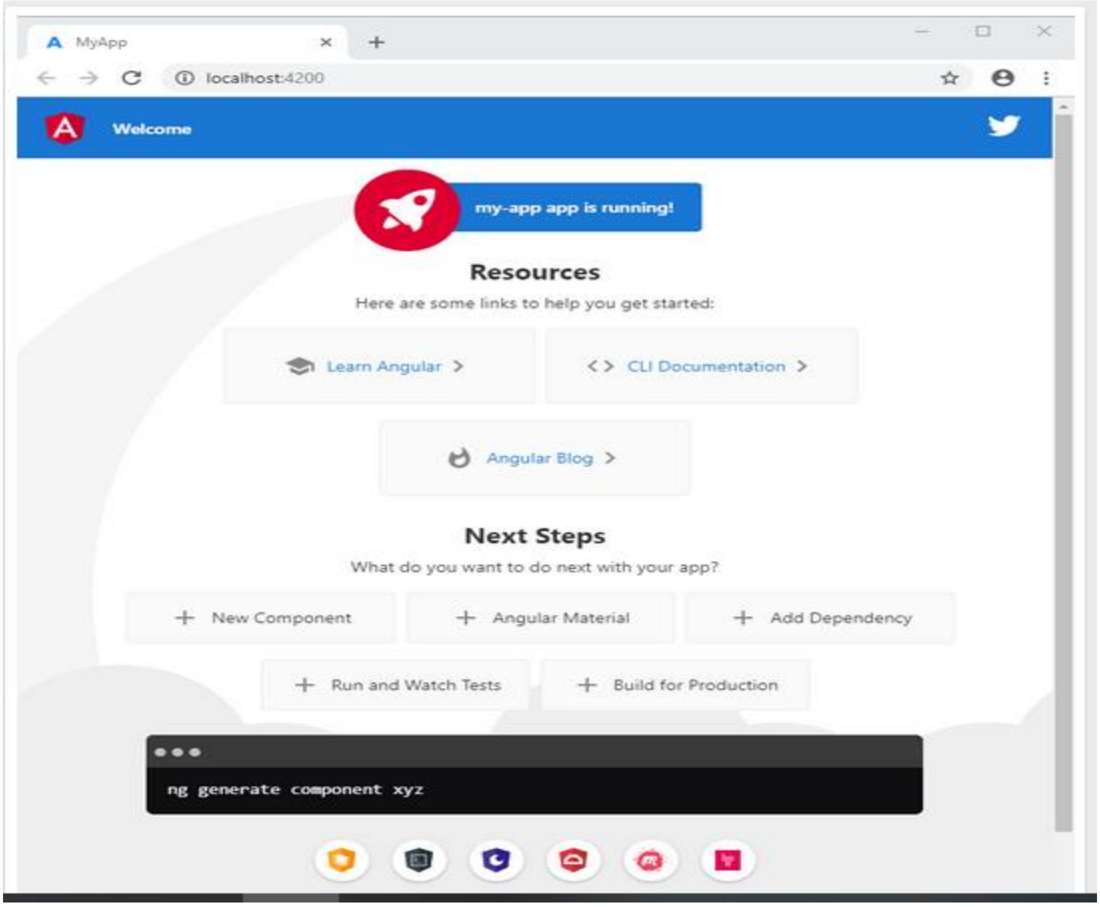
The Angular CLI includes a server, so that you can easily build and serve your app locally.

1. Go to the workspace folder (my-app).
2. Launch the server by using the CLI command ng serve, with the --open option.

```
cd my-app

ng serve --open
```

The ng serve command launches the server, watches your files, and rebuilds the app as you make changes to those files.

The --open (or just -o) option automatically opens your browser to http://localhost:4200/.



my-app app is running!

Resources

Here are some links to help you get started:

Next Steps

What do you want to do next with your app?

```
ng generate component xyz
```

PROJECT DOCUMENTATION

INTRODUCTION

Project Overview

“DREAMS” is a web application which is meant for users who are looking for online hand made products purchase and those who want to sell their creative products. The proposed system includes three users they are administrator, customer and the seller. Registered customers can login to the site and search products based on products names or by searching seller or brand names and place their orders and do transactions. Sellers have the option to create their profile, their own brands, add products and sell their products. They can view the orders, complaints, or customization messages as well from the customers. The administrator has the central control over the whole system. Administrator do the verification of seller identity, approve sellers, add product or seller categories etc.

Project Specification

The proposed system is a website in which user can buy handmade creative products as well as individuals can sell their products efficiently. We will also provide users to give customization comments to customize their products, they can view the seller or brand details, payment details etc.

The system includes 3 modules. They are:

1. **Admin Module**

The site admin have an overall control on the website. He can perform the functionalities like,

- Login to the Application
- Add/View/Update/Delete seller category/product category
- View registered sellers and registered customers
- Activate/Deactivate registered customers
- Approve/Reject Registered Sellers
- Manage Profile and password

2. Customer Module

Customer can perform functionalities like,

- Registration/Login and view/manage profile/change password
- View products category wise, subcategory wise, based on brand names, based on seller types etc.
- Search products by products name
- Sort products by brand name, subcategory name, price: high to low/low to high etc.
- Add/View/Update products in cart/wish list
- Add customization comments based on the requirement.
- View/Download order summary
- Track Order
- Add Review, Rating and Post Complaints for the purchased products
- Payment with Paytm Gateway

3. Seller Module

A Seller can manage their business effectively by this application. They can perform the functionalities like,

- Registration/Login and View/Manage profile/change password
- Create/View/Update/Delete Brands
- Manage Product Subcategory
- Add/View/Update/Delete Products in stock
- Manage customer order in different states like New Orders/Packed Orders/Dispatched orders
- View/Replay Complaints from Customer.
- Seller will receive notification for new orders, new complaints and for product running out of stock.

SYSTEM STUDY

Introduction

System analysis is a process of gathering and interpreting facts, diagnosing problems and the information to recommend improvements on the system. It is a problem solving activity that requires intensive communication between the system users and system developers. System analysis or study is an important phase of any system development process. The system is studied to the minute's detail and analyzed. The system analyst plays the role of the interrogator and dwells deep into the working of the present system. The system is viewed as a whole and the input to the system are identified. The outputs from the organizations are traced to the various processes. System analysis is concerned with becoming aware of the problem, identifying the relevant and decisional variables, analyzing and synthesizing the various factors and determining an optimal or at least a satisfactory solution or program of action.

A detailed study of the process must be made by various techniques like interviews, questionnaires etc. The data collected by these sources must be scrutinized to arrive to a conclusion. The conclusion is an understanding of how the system functions. This system is called the existing system. Now the existing system is subjected to close study and problem areas are identified. The designer now functions as a problem solver and tries to sort out the difficulties that the enterprise faces. The solutions are given as proposals. The proposal is then weighed with the existing system analytically and the best one is selected. The proposal is presented to the user for an endorsement by the user. The proposal is reviewed on user request and suitable changes are made. This is loop that ends as soon as the user is satisfied with proposal.

Preliminary study is the process of gathering and interpreting facts, using the information for further studies on the system. Preliminary study is problem solving activity that requires intensive communication between the system users and system developers. It does various feasibility studies. In these studies, a rough figure of the system activities can be obtained, from which the decision about the strategies to be followed for effective system study and analysis can be taken.

EXISTING SYSTEM

There is no automated existing system to manage home based business processes. The customers can buy products directly by meeting the seller or they can contact the sellers via phone or social media and place orders. So there is no efficient management of buying and selling of products in the existing system. The proposed system rectify the drawbacks of the present system. All the processes are automated and sellers can able to manage their stock using the website efficiently.

Drawbacks of existing system

□ No publicity for the products

□ One way and face-to-face marketing

□ Human effort is needed.

□ Customers do not get a chance of identify good products or sellers who create these products

PROPOSED SYSTEM

The proposed system is defined to meets all the disadvantages of the existing system. It is necessary to have a system that is more user friendly and user attractive for business growth. On such consideration the system is proposed. In our proposed system there is an admin who can view and manage all the registered sellers and customers. It allows customers to make their orders and do their transactions by using online payment method .Users of this proposed system are admin, customer and seller.

The aim of proposed system is to develop a system of improved facilities. The system provides proper security and reduces the manual work. Online shopping advancements have been so drastic that it has evolved to be a part of our life. Today customer doesn’t drive down to some shop for buying a product but preferably check over the internet for price, offers, reviews and order online. The existing system working procedure is like sellers create their products and find their customers via Facebook, Instagram or through any other social networking sites. Similarly customers do not get a chance to find good products or sellers from the existing method. The major drawbacks in current offline system are poor order or stock management.

ADVANTAGES OF PROPOSED SYSTEM

The system requires very low system resources and the system will work in almost all configurations. It has got following features:

- *You are able to get creative and innovative handmade products:-*

  Our customers will get creative handmade products with affordable price and they have the option to get customized product from the sellers. Also sellers can able to do proper business management and can also improve the quality and uniqueness of the products based on the market trends.

- *Better security: -*

  For data to remain secure measures must be taken to prevent unauthorized access. Security means that data are protected from various forms of destruction. The system security problem can be divided into four related issues: security, integrity, privacy and confidentiality. Username and password requirement to sign in ensures security. It will also provide data security as we are using the secured databases for maintaining the documents.

- *Ensure data accuracy: -*

  The proposed system eliminates the manual errors while entering the details of the users during the registration.

- *Better service: -*

  The product will avoid the burden of hard copy storage. We can also conserve the time and human resources for doing the same task. The data can be maintained for longer period with no loss of data.

- *User friendliness and interactive: -*

  The proposed system's interface help the users to perform their operations without any confusions or difficulties. A customer can easily find their interested products and can able to purchase it. A seller can get notification when a new order, complaint or if a product running out of stock etc and which helps the seller to immediately take an action on it.

➢*Minimum time required: -*

The customers can view products in different ways like category wise, subcategory wise, brand wise or can view by directly searching a product name which helps to purchase a product with minimum time. Similarly sellers can manage their orders and other operations easily using this website.

REQUIREMENT ANALYSIS

Feasibility study

Feasibility study is made to see if the project on completion will serve the purpose of the organization for the amount of work, effort and the time that spend on it. Feasibility study lets the developer foresee the future of the project and the usefulness. A feasibility study of a system proposal is according to its workability, which is the impact on the organization, ability to meet their user needs and effective use of resources. Thus, when a new application is proposed it normally goes through a feasibility study before it is approved for development.

The document provides the feasibility of the project that is being designed and lists various areas that were considered very carefully during the feasibility study of this project such as Technical, Economic and Operational feasibilities. The following are its features: -

Economical Feasibility

The developing system must be justified by cost and benefit. Criteria to ensure that effort is concentrated on project, which will give best, return at the earliest. One of the factors, which affect the development of a new system, is the cost it would require.

The following are some of the important financial questions asked during preliminary investigation:

- The costs conduct a full system investigation.
- The cost of the hardware and software.
- The benefits in the form of reduced costs or fewer costly errors.

The proposed system is developed as part of project work, there is no manual cost to spend for the proposed system. Also all the resources are already available, it give an indication of the system is economically possible for development.

The cost of project, DREAMS was divided according to the system used, its development cost and cost for hosting the project. According to all the calculations the project was developed in a low cost. As it is completely developed using open source software.

Technical Feasibility

The system must be evaluated from the technical point of view first. The assessment of this feasibility must be based on an outline design of the system requirement in the terms of input, output, programs and procedures. Having identified an outline system, the investigation must go on to suggest the type of equipment, required method developing the system, of running the system once it has been designed.

Technical issues raised during the investigation are:

- ➢ Does the existing technology sufficient for the suggested one?
- ➢ Can the system expand if developed?

The project should be developed such that the necessary functions and performance are achieved within the constraints. The project requires High Resolution Scanning device and utilizes Cryptographic techniques. Through the technology may become obsolete after some period of time, due to the fact that newer version of same software supports older versions, the system may still be used. So there are minimal constraints involved with this project. The system has been developed using HTML, CSS in front end and MySQL in server in back end, the project is technically feasible for development. The system has been developed using HTML, CSS, PHP and MySQL in server back end, the project is technically feasible for development. The System used was also of good performance of Processor Intel i3 core; RAM 4GB and, Hard disk 1TB

Behavioral Feasibility

The proposed system includes the following questions:

- ➢ Is there sufficient support for the users?
- ➢ Will the proposed system cause harm?

The project would be beneficial because it satisfies the objectives when developed and installed.

All behavioral aspects are considered carefully and conclude that the project is behaviorally feasible.

DREAMS, GUI is user friendly so that users can easily use it without any training.

Requirement Modeling

UML diagram

UML is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems. UML was created by the Object Management Group (OMG) and UML 1.0 specification draft was proposed to the OMG in January 1997.

UML stands for **Unified Modeling Language**. UML is different from the other common programming languages such as C++, Java, COBOL, etc. UML is a pictorial language used to make software blueprints. UML can be described as a general purpose visual modeling language to visualize, specify, construct, and document software system. Although UML is generally used to model software systems, it is not limited within this boundary. It is also used to model non-software systems as well. For example, the process flow in a manufacturing unit, etc. UML is not a programming language but tools can be used to generate code in various languages using UML diagrams. UML has a direct relation with object oriented analysis and design. After some standardization, UML has become an OMG standard. All the elements, relationships are used to make a complete UML diagram and the diagram represents a system. The visual effect of the UML diagram is the most important part of the entire process. All the other elements are used to make it complete. UML includes the following nine diagrams.

- Activity diagram
- Use case diagram
- Sequence diagram
- Collaboration diagram
- State chart diagram
- Class diagram
- Object diagram
- Component diagram
- Deployment diagram

1. UML Activity Diagram

Activity Diagram describe the dynamic aspects of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system.

The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc.

Activity diagrams are not only used for visualizing the dynamic nature of a system, but they are also used to construct the executable system by using forward and reverse engineering techniques. The only missing thing in the activity diagram is the message part.

It does not show any message flow from one activity to another. Activity diagram is sometimes considered as the flowchart. Although the diagrams look like a flowchart, they are not.

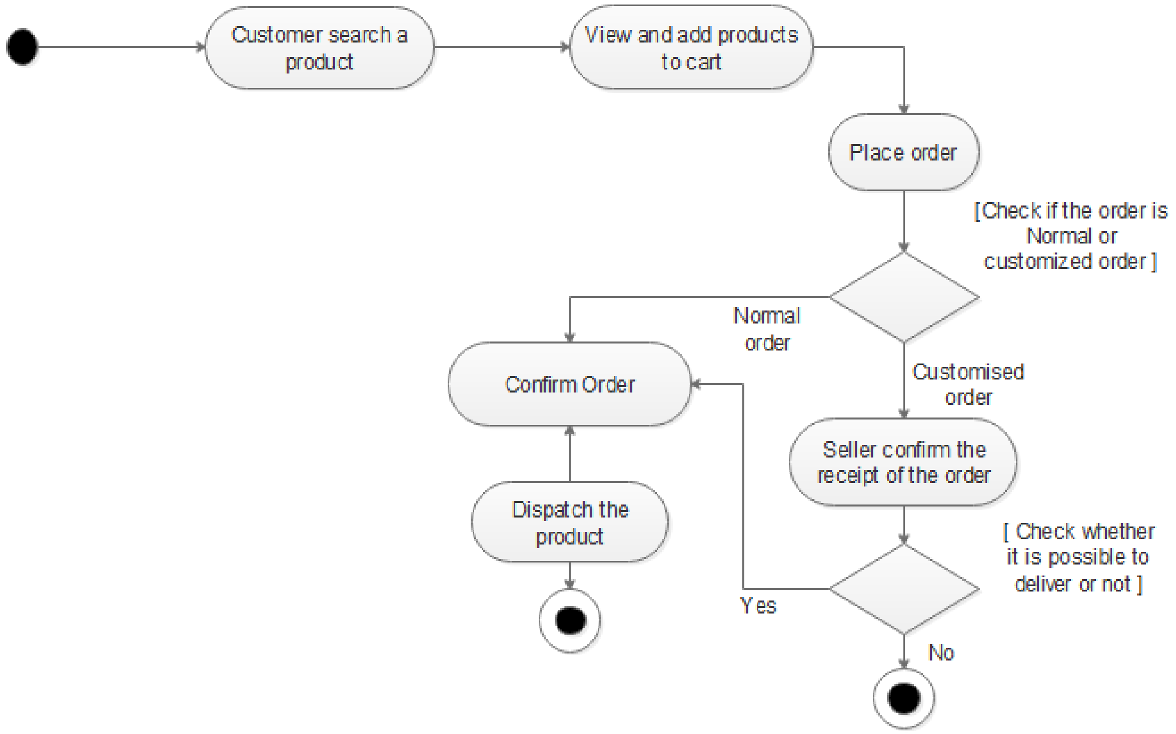

Customer search a product
View and add products to cart
Place order
[Check if the order is Normal or customized order]
Normal order
Customised order
Confirm Order
Seller confirm the receipt of the order
[Check whether it is possible to deliver or not]
Dispatch the product
Yes
No

Fig. Activity diagram for purchasing products

2. UML Use Case Diagram

A use case diagram is a graphic depiction of the interactions among the elements of a system. A use case is a methodology used in system analysis to identify, clarify, and organize system requirements. In this context, the term "system" refers to something being developed or operated, such as a mail-order product sales and service Web site. Use case diagrams are employed in UML (Unified Modeling Language), a standard notation for the modeling of real-world objects and systems.

System objectives can include planning overall requirements, validating a hardware design, testing and debugging a software product under development, creating an online help reference, or performing a consumer-service-oriented task. For example, use cases in a product sales environment would include item ordering, catalog updating, payment processing, and customer relations. A use case diagram contains four components.

- The boundary, which defines the system of interest in relation to the world around it.
- The actors, usually individuals involved with the system defined according to their roles.
- The use cases, which the specific roles are played by the actors within and around the system.
- The relationships between and among the actors and the use cases.

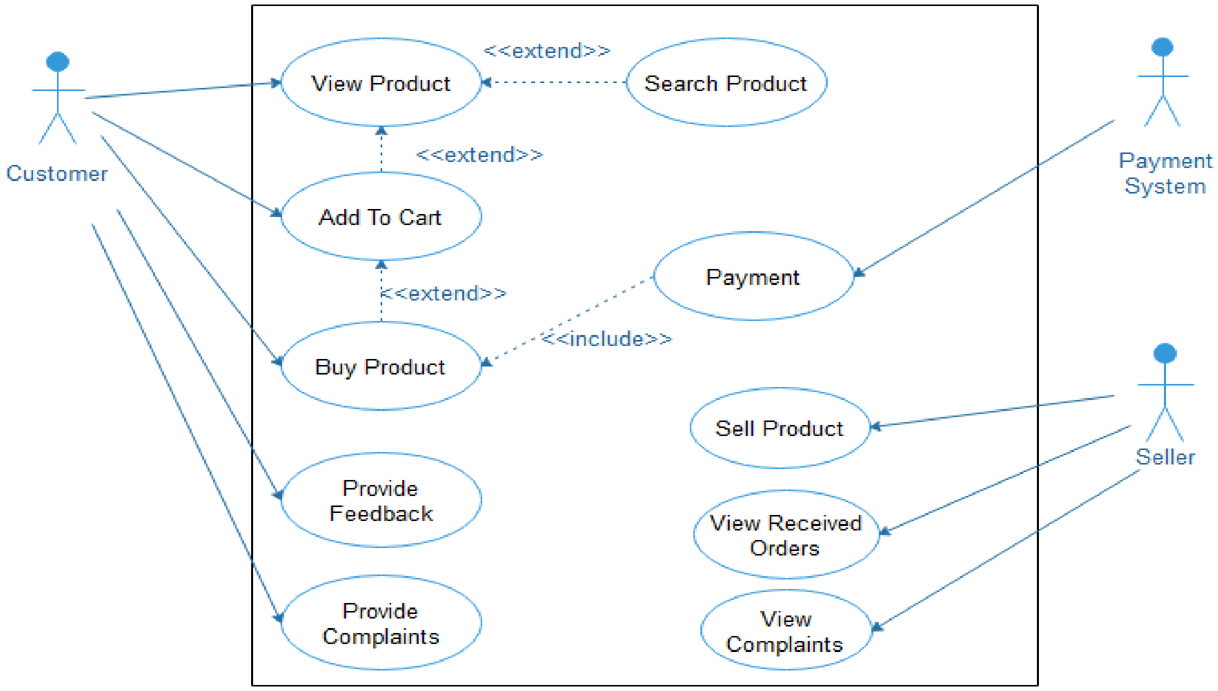


Fig. Use case diagram for online shopping

3. UML Sequence Diagram

A sequence diagram is an interaction diagram that shows how objects operate with one another and in what order. It is a construct of a message sequence chart.

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams or event scenarios. A sequence diagram shows, as parallel vertical lines (*lifelines*), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner

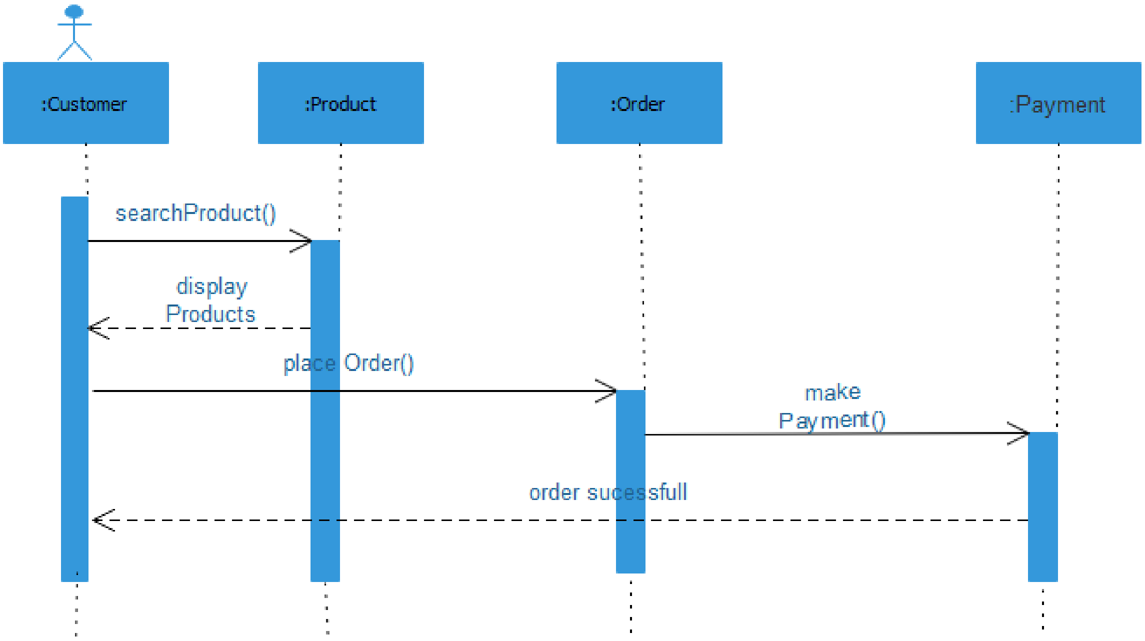


Fig. Sequence diagram for search and order product

4. UML Collaboration Diagram

The collaboration diagram is used to show the relationship between the objects in a system. Both the sequence and the collaboration diagrams represent the same information but differently. Instead of showing the flow of messages, it depicts the architecture of the object residing in the system as it is based on object-oriented programming. An object consists of several features. Multiple objects present in the system are connected to each other. The collaboration diagram, which is also known as a communication diagram, is used to portray the object's architecture in the system.

Notations of a Collaboration Diagram

Following are the components of a component diagram that are enlisted below:

1. **Objects:** The representation of an object is done by an object symbol with its name and class underlined, separated by a colon. In the collaboration diagram, objects are utilized in the following ways:
    - The object is represented by specifying their name and class.
    - It is not mandatory for every class to appear.
    - A class may constitute more than one object.
    - In the collaboration diagram, firstly, the object is created, and then its class is specified.
    - To differentiate one object from another object, it is necessary to name them.
    -

2. **Actors:** In the collaboration diagram, the actor plays the main role as it invokes the interaction. Each actor has its respective role and name. In this, one actor initiates the use case.
3. **Links:** The link is an instance of association, which associates the objects and actors. It portrays a relationship between the objects through which the messages are sent. It is represented by a solid line. The link helps an object to connect with or navigate to another object, such that the message flows are attached to links.
4. **Messages:** It is a communication between objects which carries information and includes a sequence number, so that the activity may take place. It is represented by a labeled arrow, which is placed near a link. The messages are sent from the sender to the receiver, and the

direction must be navigable in that particular direction. The receiver must understand the message.

The collaborations are used when it is essential to depict the relationship between the object. Both the sequence and collaboration diagrams represent the same information, but the way of portraying it quite different. The collaboration diagrams are best suited for analyzing use cases.

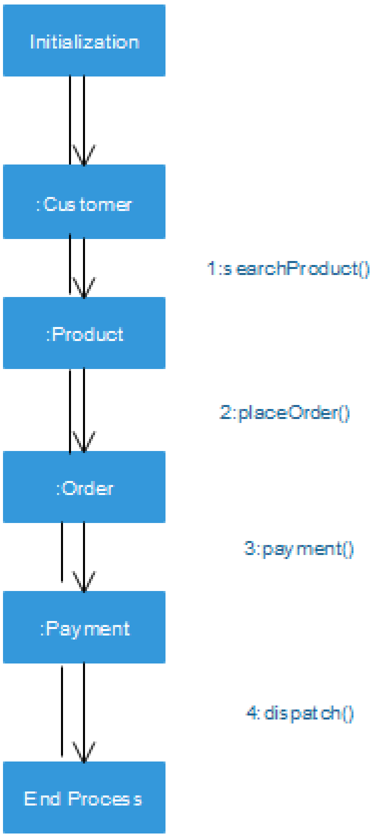


Fig. Collaboration Diagram for online shopping of products

5. UML State Chart Diagram

It describes different states of a component in a system. The states are specific to a component/object of a system. A Statechart diagram describes a state machine. State machine can be defined as a machine which defines different states of an object and these states are controlled by external or internal events. They define different states of an object during its lifetime and these states are changed by events. Statechart diagrams are useful to model the reactive systems. Reactive systems can be defined as a system that responds to external or internal events.

Statechart diagram describes the flow of control from one state to another state. States are defined as a condition in which an object exists and it changes when some event is triggered. The most important purpose of Statechart diagram is to model lifetime of an object from creation to termination. Statechart diagrams are also used for forward and reverse engineering of a system. However, the main purpose is to model the reactive system.

Following are the main purposes of using Statechart diagrams −

- To model the dynamic aspect of a system.
- To model the life time of a reactive system.
- To describe different states of an object during its life time.
- Define a state machine to model the states of an object.

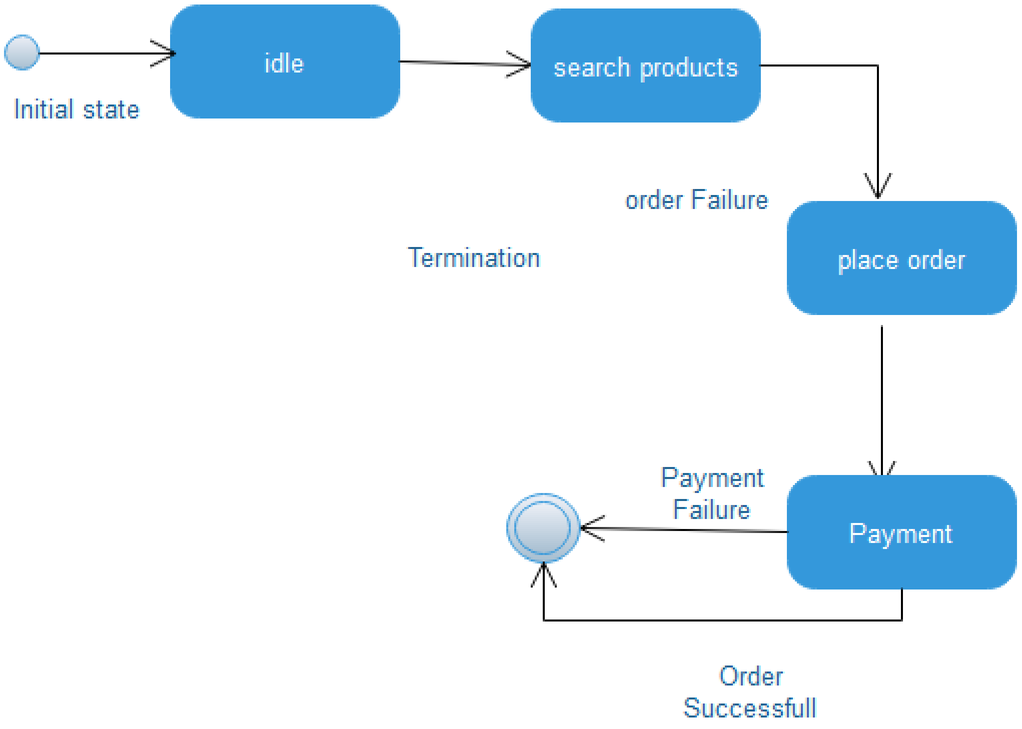


Fig. Statechart Diagram for online shopping

6. UML Class Diagram

Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application.

Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modeling of object-oriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages.

Class diagram shows a collection of classes, interfaces, associations, collaborations, and constraints. It is also known as a structural diagram.

The purpose of the class diagram can be summarized as −

- Analysis and design of the static view of an application.
- Describe responsibilities of a system.
- Base for component and deployment diagrams.
- Forward and reverse engineering.

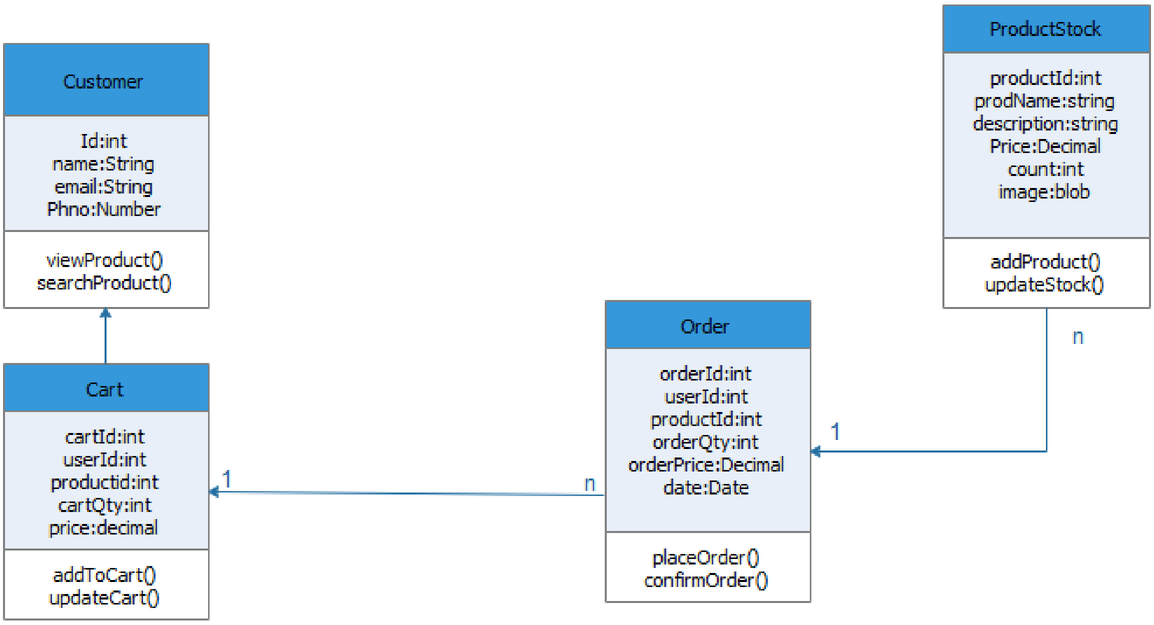


Fig. Class Diagram for online shopping site

7. UML Object Diagram

Object diagrams are derived from class diagrams so object diagrams are dependent upon class diagrams. Object diagrams represent an instance of a class diagram. The basic concepts are similar for class diagrams and object diagrams.

Object diagrams also represent the static view of a system but this static view is a snapshot of the system at a particular moment. Object diagrams are used to render a set of objects and their relationships as an instance.

The purpose of the object diagram can be summarized as −

- Forward and reverse engineering.
- Object relationships of a system
- Static view of an interaction.
- Understand object behavior and their relationship from practical perspective

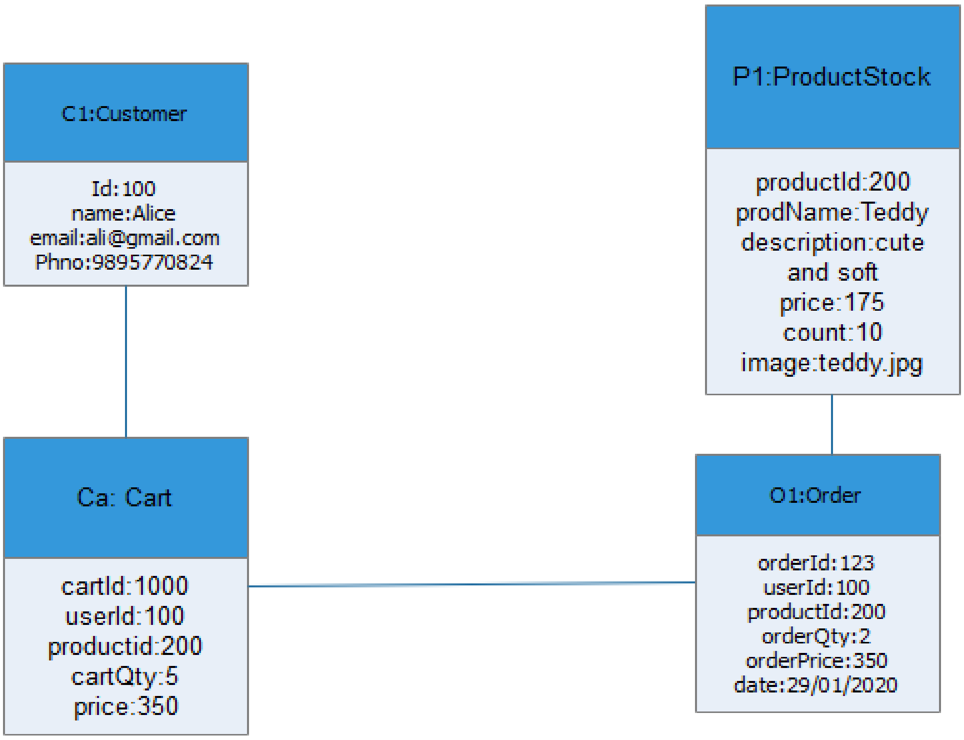


Fig. Object diagram for online shopping site

8. UML Component Diagram

Component diagram is a special kind of diagram in UML. The purpose is also different from all other diagrams discussed so far. It does not describe the functionality of the system but it describes the components used to make those functionalities. Thus from that point of view, component diagrams are used to visualize the physical components in a system. These components are libraries, packages, files, etc.

Component diagrams can also be described as a static implementation view of a system. Static implementation represents the organization of the components at a particular moment. A single component diagram cannot represent the entire system but a collection of diagrams is used to represent the whole.

The purpose of the component diagram can be summarized as −

- Visualize the components of a system.
- Construct executable by using forward and reverse engineering.
- Describe the organization and relationships of the components.

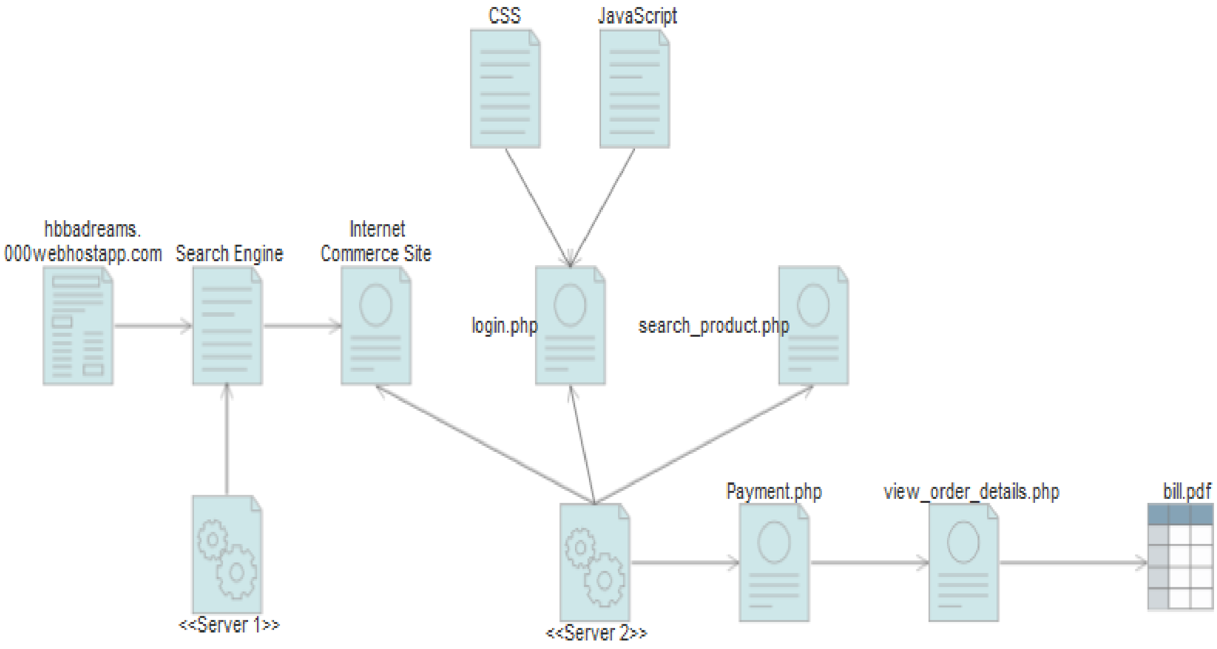


Fig. Component diagram for online shopping site

9. UML Deployment Diagram

Deployment diagrams are used to visualize the topology of the physical components of a system, where the software components are deployed. Deployment diagrams are used to describe the static deployment view of a system. Deployment diagrams consist of nodes and their relationships.

It ascertains how software is deployed on the hardware. It maps the software architecture created in design to the physical system architecture, where the software will be executed as a node. Since it involves many nodes, the relationship is shown by utilizing communication paths.

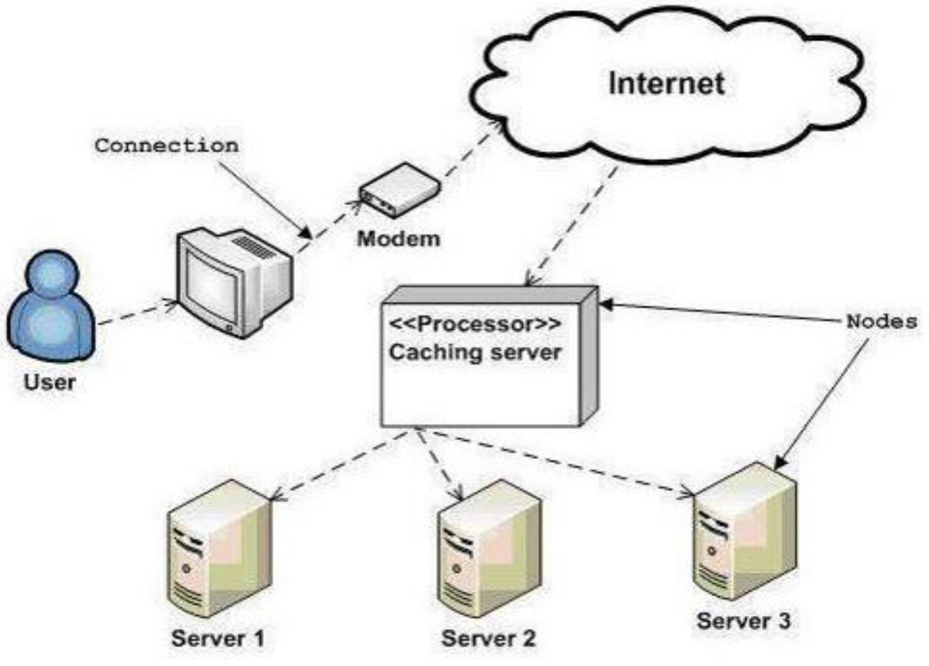


Fig. Deployment diagram

System specification

Hardware Specification

Processor	-	Intel core i3
RAM	-	4 GB
Hard disk	-	1 TB

Software Specification

Front End	-	HTML, CSS
Backend	-	MYSQL
Client on PC	-	Windows 7 and above.
Technologies used	-	JS, HTML5, AJAX, J Query, PHP, CSS

Software description

PHP

PHP is a server side scripting language designed for web development but also used as a general purpose programming language. PHP is now installed on more than 244 million websites and 2.1 million web servers. Originally created by Rasmus Ledorf in 1995, the reference implementation of PHP is now produced by the PHP group. While PHP originally stood for personal Home page ,it now stands for PHP:HypertextPreprocessor, a recursive acronym.PHP code is interpreted by a web server with a PHP processor module which generates the resulting web page.PHP commands can be embedded directly into a HTML source document rather than calling an external file to process data. It has also evolved to include a command-line interface capability and can be used in standalone incompatible with the GNU General Public License (GPL) due to restrictions on the usage of the term PHP.PHP can be deployed on most web servers and also as a standalone shell on almost every operating system and platform, free of charge.

MySQL

MySQL, the most popular Open Source SQL database management system, is developed, distributed, and supported by Oracle Corporation. The MySQL Web site provides the latest information about MySQL software.

- **MySQL is a database management system**

A database is a structured collection of data. It may be anything from a simple shopping list to a picture gallery or the vast amounts of information in a corporate network. To add, access, and process data stored in a computer database, you need a database management system such as MySQL Server. Since computers are very good at handling large amounts of data, database management systems play a central role in computing, as standalone utilities, or as parts of other applications.

- **MySQL databases are relational.**

A relational database stores data in separate tables rather than putting all the data in one big storeroom. The database structures are organized into physical files optimized for speed. The logical model, with objects such as databases, tables, views, rows, and columns, offers a flexible programming environment. You set up rules governing the relationships between different data fields, such as one-to-one, one-to-many, unique, required or optional, and "pointers" between different tables. The database enforces these rules, so that with a well-designed database, your application never sees inconsistent, duplicate, orphan, out-of-date, or missing data. The SQL part of "MySQL" stands for "Structured Query Language". SQL is the most common standardized language used to access databases. Depending on your programming environment, you might enter SQL directly (for example, to generate reports), embed SQL statements into code written in another language, or use a language-specific API that hides the SQL syntax. SQL is defined by the ANSI/ISO SQL Standard. The SQL standard has been evolving since 1986 and several versions exist. In this manual, "SQL92" refers to the standard released in 1992,"SQL: 1999" refers to the standard released in 1999, and "SQL: 2003" refers to the current version of the standard. We use the phrase "the SQL standard" to mean the current version of the SQL Standard at any time.

- **MySQL software is Open Source.**

Open Source means that it is possible for anyone to use and modify the software. Anybody can download the MySQL software from the Internet and use it without paying anything. If you wish, you may study the source code and change it to suit your needs. The MySQL software uses the GPL (GNU General Public License), to define what you may and may not do with the software in different situations. If you feel uncomfortable with the GPL or need to embed MySQL code into a commercial application, you can buy a commercially licensed version from us. See the MySQL Licensing Overview for more information.

- **The MySQL Database Server is very fast, reliable, scalable, and easy to use.**

If that is what you are looking for, you should give it a try. MySQL Server can run comfortably on a desktop or laptop, alongside your other applications, web servers, and so on, requiring little or no attention. If you dedicate an entire machine to MySQL, you can adjust the settings to take advantage of all the memory, CPU power, and I/O capacity available.

- **MySQL Server works in client/server or embedded systems.**

The MySQL Database Software is a client/server system that consists of a multi-threaded SQL server that supports different back ends, several different client programs and libraries, administrative tools, and a wide range of application programming interfaces (APIs). We also provide MySQL Server as an embedded multi-threaded library that you can link into your application to get a smaller, faster, easier-to-manage standalone product.

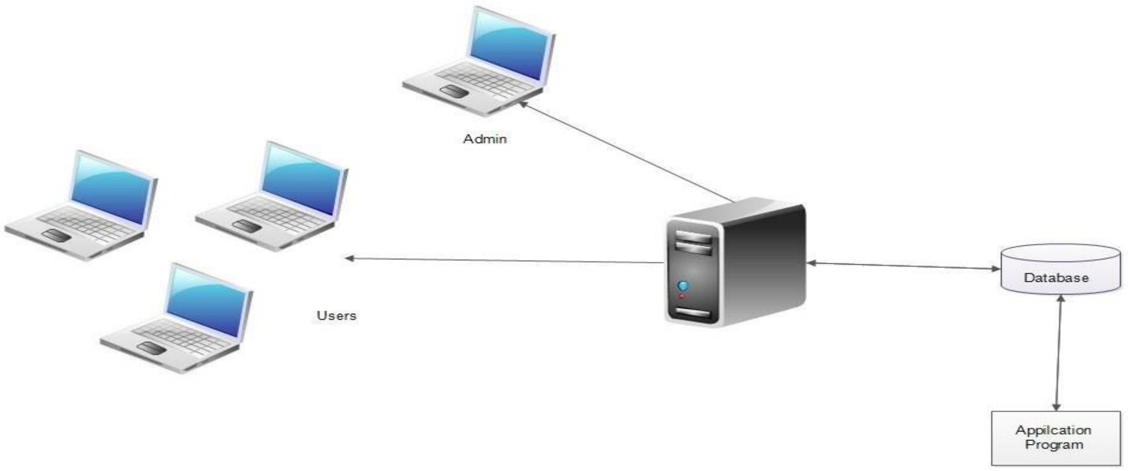
- **A large amount of contributed MySQL software is available.**

MySQL Server has a practical set of features developed in close cooperation with our users. It is very likely that your favorite application or language supports the MySQL Database Server.

System design

Design is the first step into the development phase for any engineered product or system. Design is a creative process. A good design is the key to effective system. The term "design" is defined as "the process of applying various techniques and principles for the purpose of defining a process or a system in sufficient detail to permit its physical realization". It may be defined as a process of applying various techniques and principles for the purpose of defining a device, a process or a system in sufficient detail to permit its physical realization. Software design sits at the technical kernel of the software engineering process and is applied regardless of the development paradigm that is used. The system design develops the architectural detail required to build a system or product. As in the case of any systematic approach, this software too has undergone the best possible design phase fine tuning all efficiency, performance and accuracy levels. The design phase is a transition from a user oriented document to a document to the programmers or database personnel. System design goes through two phases of development: Logical and Physical Design.

Architectural Design



The registered customer, seller, and admin can accesses the website through internet using their Laptop, Smart Phone, Tablet or Desktop Computer. The System's application program processes the user's request and provides the required services by taking data from the system database.

Module Design

Admin Module

The administrator of the company is allowed to access all the services in the system.
Admin has the overall control of the system. Admin can add or update product categories, seller categories etc. Admin can View all the registered sellers, can able to approve or reject sellers and also can able to view all registered customer details.

Manage user details, Add seller types and product categories	Deactivate/Activate the registered users.
Approve/Reject Registered Sellers	View users feedback

Registered Customer Module

After registration, customers can view products, search products, can add products to cart, purchase products through secure online payment.

Customer registration, login	Search products/Add to cart/Add to Wish list
Checkout and Order products	View Order Details/Order summary
Track Ordered products	Add Review, Rating and Complaints

Registered Seller Module

After registration, customers can view products, search products, can add products to cart, purchase products through secure online payment.

Seller registration, login	Create/View/Update Brands
Add/view/update subcategory or products	View/update customer order
View/Replay to Complaints from customer	Manage profile and Product Stock

Database Design

A database is an organized mechanism that has the capability of storing information through which a user can retrieve stored information in an effective and efficient manner. The data is the purpose of any database and must be protected.

The database design is a two level process. In the first step, user requirements are gathered together and a database is designed which will meet these requirements as clearly as possible. This step is called Information Level Design and it is taken independent of any individual DBMS.

In the second step, this Information level design is transferred into a design for the specific DBMS that will be used to implement the system in question. This step is called Physical Level Design, concerned with the characteristics of the specific DBMS that will be used. A database design runs parallel with the system design. The organization of the data in the database is aimed to achieve the following two major objectives.

- Data Integrity
- Data independence

Relational Database Management System (RDBMS)

A relational model represents the database as a collection of relations. Each relation resembles a table of values or file of records. In formal relational model terminology, a row is called a tuple, a column header is called an attribute and the table is called a relation. A relational database consists of a collection of tables, each of which is assigned a unique name. A row in a tale represents a set of related values.

Relations, Domains & Attributes

A table is a relation. The rows in a table are called tuples. A tuple is an ordered set of n elements. Columns are referred to as attributes. Relationships have been set between every table in the database. This ensures both Referential and Entity Relationship Integrity. A domain D is a set of atomic values. A common method of specifying a domain is to specify a data type from which the data values forming the domain are drawn. It is also useful to specify a name for the domain to help in interpreting its values.

Every value in a relation is atomic, that is not decomposable.

Relationships

- Table relationships are established using Key. The two main keys of prime importance are Primary Key & Foreign Key. Entity Integrity and Referential Integrity Relationships can be established with these keys.
- Entity Integrity enforces that no Primary Key can have null values.
- Referential Integrity enforces that no Primary Key can have null values.
- Referential Integrity for each distinct Foreign Key value, there must exist a matching Primary Key value in the same domain. Other key are Super Key and Candidate Keys.

Normalization

Data are grouped together in the simplest way so that later changes can be made with minimum impact on data structures. Normalization is formal process of data structures in manners that eliminates redundancy and promotes integrity. Normalization is a technique of separating redundant fields and breaking up a large table into a smaller one. It is also used to avoid insertion, deletion, and updating anomalies.

Normal form in data modelling use two concepts, keys and relationships. A key uniquely identifies a row in a table.

There are two types of keys, primary key and foreign key. A primary key is an element or a combination of elements in a table whose purpose is to identify records from the same table. A foreign key is a column in a table that uniquely identifies record from a different table. All the tables have been normalized up to the third normal form.

As the name implies, it denotes putting things in the normal form. The application developer via normalization tries to achieve a sensible organization of data into proper tables and columns and where names can be easily correlated to the data by the user. Normalization eliminates repeating groups at data and thereby avoids data redundancy which proves to be a great burden on the computer resources. These include:

- ✓ Normalize the data.
- ✓ Choose proper names for the tables and columns.
- ✓ Choose the proper name for the data.

First Normal Form

The First Normal Form states that the domain of an attribute must include only atomic values and that the value of any attribute in a tuple must be a single value from the domain of that attribute. In other words 1NF disallows “relations within relations” or “relations as attribute values within tuples”. The only attribute values permitted by 1NF are single atomic or indivisible values. The first step is to put the data into First Normal Form. This can be donor by moving data into separate tables where the data is of similar type in each table. Each table is given a Primary Key or Foreign Key as per requirement of the project. In this we form new relations for each non-atomic attribute or nested relation. This eliminated repeating groups of data. A relation is said to be in first normal form if only if it satisfies the constraints that contain the primary key only.

Second Normal Form

According to Second Normal Form, for relations where primary key contains multiple attributes, no non-key attribute should be functionally dependent on a part of the primary key. In this we decompose and setup a new relation for each partial key with its dependent attributes. Make sure to keep a relation with the original primary key and any attributes that are fully functionally dependent on it. This step helps in taking out data that is only dependent on a part of the key.

A relation is said to be in second normal form if and only if it satisfies all the first normal form conditions for the primary key and every non-primary key attributes of the relation is fully dependent on its primary key alone.

Third Normal Form

According to Third Normal Form, Relation should not have a non-key attribute functionally determined by another non-key attribute or by a set of non-key attributes. That is, there should be no transitive dependency on the primary key. In this we decompose and set up relation that includes the non-key attributes that functionally determines other non-key attributes. This step is taken to get rid of anything that does not depend entirely on the Primary Key. A relation is said to be in third normal form if only if it is in second normal form and more over the non key attributes of the relation should not be depend on other non-key attribute.

TABLES

Table No 01 : tbl_user_type

Primary Key : user_type_id

Foreign Key :

Field Name	Type	Size	Description
user_type_id	Int	10	Primary key of user_type table
type_name	Varchar	20	To store user types

Table No 02 : tbl_state

Primary Key : state_id

Foreign Key :

Field Name	Type	Size	Description
state_id	Int	10	Primary key of state table
state_name	Varchar	20	To store state names
Status	Varchar	10	Status

Table No 03 : tbl_district

Primary Key : district_id

Foreign Key : state_id

Field Name	Type	Size	Description
district_id	Int	10	Primary key of District table
state_id	int	10	Foreign key of state table
district_name	Varchar	30	To store district names

Table No 04 : tbl_registration

Primary Key : register_email

Foreign Key : user_type_id

Field Name	Type	Size	Description
register_email	Varchar	50	Primary key of registration table
user_type_id	Int	10	Foreign key of user_type table
Fname	Varchar	30	First name
Lname	Varchar	30	Last name
mobile_num	Varchar	15	Mobile number
Status	Varchar	10	To check whether user is available or not

Table No 05 : tbl_login

Primary Key : login_id

Foreign Key : register_email

Field Name	Type	Size	Description
login_id	Int	10	Primary key of login table
register_email	Varchar	50	Foreign key of user_type table
password	Varchar	30	Store password of the user

Table No 06 : tbl_seller_type

Primary Key : seller_type_id

Foreign Key :

Field Name	Type	Size	Description
seller_type_id	Int	10	Primary key of seller_type table
seller_type_name	Varchar	30	To store seller_type names
Status	Varchar	10	Status

Table No 07 : tbl_seller_profile_brand

Primary Key : seller_profile_brand_id

Foreign Key : seller_type_id, register_email

Field Name	Type	Size	Description
seller_profile_brand_id	Int	10	Primary key of seller_profile_brand table
seller_type_id	Int	10	Foreign key of seller_type table
register_email	Varchar	50	Foreign key of register table
brand_name	varchar	30	Unique brand name
Profile_description	Varchar	200	Description about the brand
profile_photo	Varchar	300	To store brand logo
achievements	Varchar	100	To store achievements
status	varchar	10	Status

Table No 08 : tbl_product_category

Primary Key : product_category_id

Foreign Key :

Field Name	Type	Size	Description
product_category_id	Int	10	Primary key of product_category table
category_name	varchar	30	Category name
Status	Varchar	10	status

Table No 09 : tbl_prod_subcatg

Primary Key : prod_subcatg_id

Foreign Key : product_category_id

Field Name	Type	Size	Description
prod_subcatg_id	Int	10	Primary key of prod_subcatg table
product_category_id	Int	10	Foreign key of product_category table
subcatg_name	varchar	30	Sub category name
Status	Varchar	10	Status

Table No 10 : tbl_seller_stock

Primary Key : stock_product_id

Foreign Key : seller_profile_brand_id, prod_subcatg_id

Field Name	Type	Size	Description
stock_product_id	Int	10	Primary key of stock_product table

seller_profile_brand_id	Int	10	Foreign key of seller_profile_brand table
prod_subcatg_id	Int	10	Foreign key of prod_subcatg table
product_name	Varchar	50	Product name
product_description	Varchar	100	Product description
item_price	Decimal		Item price
stock_item_count	Int	10	Stock item count
Img1	Varchar	300	Image 1 of the item
Img2	Varchar	300	Image 2 of the item
Img3	Varchar	300	Image 3 of the item
Added_date	Date		Product added date
Updated date	Date		Product info modified date
Status	Varchar	10	status

Table No 11 : tbl_customer_cart

Primary Key : customer_cart_item_id

Foreign Key : register_email,seller_stock_product_id

Field Name	Type	Size	Description
customer_cart_item_id	Int	10	Primary key of customer_cart table
register_email	varchar	30	Foreign key of register table
seller_stock_product_id	Int	10	Foreign key of seller_stock_product table

Quantity	Int	10	Number of quantity added to cart
Total price	decimal		Total Price of each item

Table No 12 : tbl_complaints

Primary Key : complaints_id

Foreign Key : register_email,seller_profile_brand_id

Field Name	Type	Size	Description
complaints_id	Int	10	Primary key of complaints table
Register_email	Varchar	30	Foreign key of register table
seller_profile_brand_id	Int	10	Foreign key of seller_profile_brand table
Complaint	Varchar	100	

Table No 13 :tbl_customer_order

Primary Key :customer_order_id

Foreign Key :register_email,user_address_id,
Stock_product_id,Delv_adres_id

Field Name	Type	Size	Description
customer_order_id	Int	10	Primary key of customer_order table
Register_email	Varchar	50	Foreign key of registration table table
Stock_product_id	int	10	Foreign key of seller product stock table

Delv_adres_id	Int	10	Foreign key of delivery address table
Purchase_qty	Int	10	Count of each item
Purchase_price	Decimal		Total price of each item
order_date	Date		Order date
Delivery_date	Date		Expected delivery date
Customize_comments	Varchar	300	Comments to customize the product
status	Varchar	10	status

Table No 14 : tbl_customer_delv_address

Primary Key : delv_adres_id

Foreign Key : register_email

Field Name	Type	Size	Description
delv_adres_id	Int	10	Primary key
Register_email	Varchar	50	Foreign key of register table
Name	Varchar	40	Foreign key of seller_profile_brand table
Mobile number	Varchar	20	Mobile number
address_line	Varchar	50	Store house /building name or number
Landmark	Varchar	50	Landmark
town_city	Varchar	30	Town

pin_code	Varchar	10	Pin code
address_type	Varchar	10	Address type like home,office etc
Status	Varchar	10	Status

Table No 15 **: tbl_seller_address**

Primary Key : seller_address_id

Foreign Key : register_email,district_id

Field Name	Type	Size	Description
Seller_address_id	Int	10	Primary key
Register_email	Varchar	50	Foreign key of registration table
District_id	int	10	Foreign key of district table
Place	Varchar	50	Place
Pincode	Varchar	15	Pincode
Status	Varchar	10	Status

Table No 16 **: tbl_seller_identity_proof**

Primary Key : seller_identity_proof_id

Foreign Key : register_email

Field Name	Type	Size	Description
Seller_address_id	Int	10	Primary key
Register_email	Varchar	50	Foreign key of registration table

Adhar_no	Varchar	20	Aadhar number
Id_proof_pdf	Varchar	250	Identity proof pdf
Profile_pic	Varchar	250	Profile picture
Status	Varchar	10	Status

Table No 17 : tbl_customer_customisation

Primary Key : customise_id

Foreign Key : register_email,order_id

Field Name	**Type**	**Size**	**Description**
customize_id	Int	10	Primary key
register_email	Varchar	50	Foreign key of registration table
order_id	Int	10	Foreign key of order table
requirements	Varchar	250	Requirements
status	Varchar	10	Status

System testing

Introduction

Software Testing is the process of executing software in a controlled manner, in order to answer the question - Does the software behave as specified? Software testing is often used in association with the terms verification and validation. Validation is the checking or testing of items, includes software, for conformance and consistency with an associated specification. Software testing is just one kind of verification, which also uses techniques such as reviews, analysis, inspections, and walkthroughs. Validation is the process of checking that what has been specified is what the user actually wanted.

Other activities which are often associated with software testing are static analysis and dynamic analysis. Static analysis investigates the source code of software, looking for problems and gathering metrics without actually executing the code. Dynamic analysis looks at the behavior of software while it is executing, to provide information such as execution traces, timing profiles, and test coverage information.

Testing is a set of activity that can be planned in advanced and conducted systematically. Testing begins at the module level and work towards the integration of entire computers based system. Nothing is complete without testing, as it vital success of the system testing objectives, there are several rules that can serve as testing objectives. They are:

Testing is a process of executing a program with the intent of finding an error.

- A good test case is one that has high possibility of finding an undiscovered error.
- A successful test is one that uncovers an undiscovered error.

If a testing is conducted successfully according to the objectives as stated above, it would uncover errors in the software. Also testing demonstrate that the software function appear to be working according to the specification, that performance requirement appear to have been met.

There are three ways to test program.

- For correctness
- For implementation efficiency
- For computational complexity

Test for correctness are supposed to verify that a program does exactly what it was designed to do. This is much more difficult than it may at first appear, especially for large programs.

Test Plan

A test plan implies a series of desired course of action to be followed in accomplishing various testing methods. The Test Plan acts as a blue print for the action that is to be followed. The software engineers create a computer program, its documentation and related data structures. The software developers is always responsible for testing the individual units of the programs, ensuring that each performs the function for which it was designed. There is an independent test group (ITG) which is to remove the inherent problems associated with letting the builder to test the thing that has been built. The specific objectives of testing should be stated in measurable terms. So that the mean time to failure, the cost to find and fix the defects, remaining defect density or frequency of occurrence and test work-hours per regression test all should be stated within the test plan.

The levels of testing include:

- ❖ Unit testing
- ❖ Integration Testing
- ❖ Data validation Testing
- ❖ Output Testing

Unit Testing

Unit testing focuses verification effort on the smallest unit of software design – the software component or module. Using the component level design description as a guide, important control paths are tested to uncover errors within the boundary of the module. The relative complexity of tests and uncovered scope established for unit testing. The unit testing is white-box oriented, and step can be conducted in parallel for multiple components. The modular interface is tested to ensure that information properly flows into and out of the program unit under test. The local data structure is examined to ensure that data stored temporarily maintains its integrity during all steps in an algorithm’s execution. Boundary conditions are tested to ensure that all statements in a module have been executed at least once. Finally, all error handling paths are tested.

Tests of data flow across a module interface are required before any other test is initiated. If data do not enter and exit properly, all other tests are moot. Selective testing of execution paths is an essential task during the unit test. Good design dictates that error conditions be anticipated and error handling paths set up to reroute or cleanly terminate processing when an error does occur. Boundary testing is the last task of unit testing step. Software often fails at its boundaries.

Integration Testing

Integration testing is systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with interfacing. The objective is to take unit tested components and build a program structure that has been dictated by design. The entire program is tested as whole. Correction is difficult because isolation of causes is complicated by vast expanse of entire program. Once these errors are corrected, new ones appear and the process continues in a seemingly endless loop. After performing unit testing in the System all the modules were integrated to test for any inconsistencies in the interfaces. Moreover differences in program structures were removed and a unique program structure was evolved.

Validation Testing or System Testing

This is the final step in testing. In this the entire system was tested as a whole with all forms, code, modules and class modules. This form of testing is popularly known as Black Box testing or System tests.

Black Box testing method focuses on the functional requirements of the software. That is, Black Box testing enables the software engineer to derive sets of input conditions that will fully exercise all functional requirements for a program.

Black Box testing attempts to find errors in the following categories; incorrect or missing functions, interface errors, errors in data structures or external data access, performance errors and initialization errors and termination errors.

Output Testing or User Acceptance Testing

The system considered is tested for user acceptance; here it should satisfy the firm's need. The software should keep in touch with perspective system; user at the time of developing and making changes whenever required. This done with respect to the following points:

- Input Screen Designs,
- Output Screen Designs,

The above testing is done taking various kinds of test data. Preparation of test data plays a vital role in the system testing. After preparing the test data, the system under study is tested using that test data. While testing the system by which test data errors are again uncovered and corrected by using above testing steps and corrections are also noted for future use.

Test Case

Test Case 1					
Project Name: **Dreams-A Home Based Business Application**					
Login Test Case					
Test Case ID: Fun_1			**Test Designed By:** **JILCY J ABRAHAM**		
Test Priority(Low/Medium/High): High			**Test Designed Date:** 22-04-2020		
Module Name:Login Screen			**Test Executed By :** **Ms. JETTY BENJAMIN**		
Test Title : Verify login with valid username and password			**Test Execution Date:** 25-04-2020		
Description: Test the Login Page					
Pre-Condition : User has valid user name and password					
Step	**Test Step**	**Test Data**	**Expected Result**	**Actual Result**	**Status(Pass/Fail)**
1	Navigation to Login Page		Login Page should be	Login page displayed	Pass
2	Provide Valid User name	User Name: admin@gmail.com	User should be able to Login	User Logged in and navigated to Admin Dashboard with records	Pass
3	Provide Valid Password	Password: Admin123*			
4	Click on Login button				
5	Provide Invalid User Name or password	User Name: 123@gmail Password: admin	User should not be able to Login	Message for enter valid user or password displayed	Pass
6	Provide Null User Name or Password	User Name:null Password: null			
7	Click on Login button				
Post-Condition: User is validated with database and successfully login into account. The Account session details are logged in database					

Test Case 2					
Project Name: Dreams-A Home Based Business Application					
User Registration Test Case					
Test Case ID: Fun_2			**Test Designed By: JILCY J ABRAHAM**		
Test Priority(Low/Medium/High): Medium			**Test Designed Date:** 22-04-2020		
Module Name:User Registration			**Test Executed By : Ms. JETTY BENJAMIN**		
Test Title : Verify new user registration			**Test Execution Date:** 25-04-2020		
Description: Test the User registration Page					
Pre-Condition : User should not be already registered					
Step	**Test Step**	**Test Data**	**Expected Result**	**Actual Result**	**Status(Pass/Fail)**
1	Navigation to Registration Page		Registration Page should be displayed	Registration page displayed	Pass
2	Provide null information	Name: null	Validation message should be displayed	*Mandatory field message displayed	Pass
3	Provide Valid Details of user	All the mandatory registration details of the user	User should be able to register successfully	Registration successful and home page displayed	Pass
4	Click on Register button				
Post-Condition: User is validated with database and successfully registered in the application. The Account session details are logged in database					

Implementation

Implementation is the stage of the project where the theoretical design is turned into a working system. It can be considered to be the most crucial stage in achieving a successful new system gaining the users confidence that the new system will work and will be effective and accurate. It is primarily concerned with user training and documentation. Conversion usually takes place about the same time the user is being trained or later. Implementation simply means convening a new system design into operation, which is the process of converting a new revised system design into an operational one.

At this stage the main work load, the greatest upheaval and the major impact on the existing system shifts to the user department. If the implementation is not carefully planned or controlled, it can create chaos and confusion.

Implementation includes all those activities that take place to convert from the existing system to the new system. The new system may be a totally new, replacing an existing manual or automated system or it may be a modification to an existing system. Proper implementation is essential to provide a reliable system to meet organization requirements. The process of putting the developed system in actual use is called system implementation. This includes all those activities that take place to convert from the old system to the new system. The system can be implemented only after through testing is done and if it is found to be working according to the specifications. The system personnel check the feasibility of the system. The more complex the system being implemented, the more involved will be the system analysis and design effort required to implement the three main aspects: education and training, system testing and changeover.

The implementation state involves the following tasks:

- Careful planning.
- Investigation of system and constraints.
- Design of methods to achieve the changeover.

Implementation Procedures

Implementation of software refers to the final installation of the package in its real environment, to the satisfaction of the intended uses and the operation of the system. In many organizations someone who will not be operating it, will commission the software development project. In the initial stage people doubt about the software but we have to ensure that the resistance does not build up, as one has to make sure that:

- The active user must be aware of the benefits of using the new system.
- Their confidence in the software is built up.
- Proper guidance is imparted to the user so that he is comfortable in using the application.

Before going ahead and viewing the system, the user must know that for viewing the result, the server program should be running in the server. If the server object is not up running on the server, the actual process won’t take place.

User Training

User training is designed to prepare the user for testing and converting the system. To achieve the objective and benefits expected from computer based system, it is essential for the people who will be involved to be confident of their role in the new system. As system becomes more complex, the need for training is more important. By user training the user comes to know how to enter data, respond to error messages, interrogate the database and call up routine that will produce reports and perform other necessary functions.

Operational Document

After providing the necessary basic training on computer awareness the user will have to be trained on the new application software. This will give the underlying philosophy of the use of the new system such as the screen flow, screen design type of help on the screen, type of errors while entering the data, the corresponding validation check at each entry and the ways to correct the date entered. It should then cover information needed by the specific user/ group to use the system or part of the system while imparting the training of the program on the application. This training may be different across different user groups and across different levels of hierarchy.

System Maintenance

Maintenance is the enigma of system development. The maintenance phase of the software cycle is the time in which a software product performs useful work. After a system is successfully implemented, it should be maintained in a proper manner. System maintenance is an important aspect in the software development life cycle. The need for system maintenance is for it to make adaptable to the changes in the system environment. Software maintenance is of course, far more than "Finding Mistakes".

Conclusion and Future Scope

Future Enhancement

- Develop a Mobile Application for the website
- Provide option to add videos of products, so that customers will get better idea about the product quality
- Include an option for public to post advertisement with affordable price in this website based on the approval from site admin.
- Provide more security.

Conclusion

The current system working technology is old fashioned and there is no usage of commonly used technologies like internet, digital money. The proposed system introduces facility for customer to place orders and track orders. Provides lots of advantages like viewing seller contact information, customize orders, enhanced user interface, payment options, delivery options, order process estimate, order status and many more.

Bibliography

Books/References:

- Gary B. Shelly, Harry J. Rosenblatt, “*System Analysis and Design*”, 2009.
- Roger S Pressman, “*Software Engineering*”, 1994.
- PankajJalote, “So*ftware engineering*: a precise approach”, 2006.
- James lee and Brent ware Addison, “Open source web development with LAMP”, 2003
- IEEE Std 1016 Recommended Practice for Software Design Descriptions.

Websites:

- www.w3schools.com
- www.jquery.com
- http://homepages.dcc.ufmg.br/~rodolfo/es-1-03/IEEE-Std-830-1998.pdf
- www.agilemodeling.com/artifacts/useCaseDiagram.html

APPENDIX

SAMPLE CODE

Connection code

```
<?php
$con=mysqli_connect("localhost","root","","db_dreams_project")or die ("Couldn't connect");
?>
```

Admin approve seller code

```
<?php
                include 'new_connection.php';
                $register_email=$_GET['register_email'];
                mysqli_query($con,"UPDATE  `tbl_registration` SET status='APPROVED'
WHERE register_email='$register_email'");
$from = $mail;
$to = $register_email;
$subject = "Regarding Your Dreams website registration status ";
$txt ="Dear Seller,\nYour Request for registering with Dreams Home Based Bussiness
Application is APPROVED by the Dreams Administrative Team."."\nYou can start your
Business career with Dreams Now.\nNote : At any time Dremas admin can deactivate your
account if there is any valid complaints received from the customers.\nBe a resposible
seller.\nAll the Best for your Business with Dreams.\nRegards,\nTeam Dreams";
$headers = "From:" . $from;
"CC:".$register_email;
mail($to,$subject,$txt,$headers);
 header('location:admin_reg_sellers.php');
   ?>
```

Order Payment Code

```
<?php
@ob_start();
session_start();
$mail=$_SESSION['alogin'];
$user_type_id=$_SESSION['user_type'];
$delvr_address_id=$_SESSION['delv_address_id'];
$cart_tl=$_SESSION['cart_total'];
//$con=mysqli_connect("localhost","root","","db_dreams_project")or die ("Couldn't connect");

include 'new_connection.php';
if(isset($_SESSION['alogin'])){
$mail=$_SESSION['alogin'];?>
<?php
if(isset($_POST["payment"]))
{
$cnumber=0;

$security_code=0;
$security_code=$_POST["security_code"];
$_SESSION['security_code']=$security_code;
$balance_amount=0;
$status='VALID';
$new_balance=0;
$d_bank_info=mysqli_query($con,"SELECT * FROM `tbl_bank_info` WHERE
CVV='$security_code' and status='VALID' and user_type_id='$user_type_id'");
$rowcount = mysqli_num_rows($d_bank_info);

    if($rowcount==0)
    {echo "<script>alert('Wrong CVV !Enter correct CVV');
       window.location='payment.php';</script>";
    }
    else if ($rowcount==1)
                          {

    $otp_num=rand(1000,9999);
$from = $mail;
$to = "jilcyjabraham@mca.ajce.in";
```

```
        $subject = $otp_num ." is the OTP for your  Credit Card xx9876";
                        $txt ="Dear Customer, \n ".$otp_num." is the One Time Password (OTP) for
your transaction of INR  ".$cart_tl.".00 at Dreams online on your Credit Card ending with 9876.
\nPlease Note:\n This OTP will be valid for this transaction only. \n\n If you have entered the OTP
incorrectly more than 3 times, your Credit Card will get temporarily blocked for online transactions .
\n \n Do not share your OTP with anyone. The Bank never calls to verify OTP.\n Regards,\n Team
ABCD Bank";
$headers = "From:" . $from;
"CC: somebodyelse@example.com";
mail($to,$subject,$txt,$headers);
                        $sql_otp=mysqli_query($con,"UPDATE `tbl_bank_info` SET
`temp_otp`=$otp_num WHERE CVV='$security_code' and status='VALID' and
user_type_id='$user_type_id' ");                                        }}?>


        <!DOCTYPE html>
        <html>
        <head>
        <title>Dreams|Customer Home</title>
        <!-- for-mobile-apps -->
        <meta name="viewport" content="width=device-width, initial-scale=1">
        <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
        <meta name="keywords" content="Grocery Store Responsive web template, Bootstrap Web
        Templates, Flat Web Templates, Android Compatible web template,
        Smartphone Compatible web template, free webdesigns for Nokia, Samsung, LG,
        SonyEricsson, Motorola web design" />
        <script type="application/x-javascript"> addEventListener("load", function() {
        setTimeout(hideURLbar, 0); }, false);
                        function hideURLbar(){ window.scrollTo(0,1); } </script>
        <!-- //for-mobile-apps -->
        <link href="css/bootstrap.css" rel="stylesheet" type="text/css" media="all" />
        <link href="css/style.css" rel="stylesheet" type="text/css" media="all" />
        <!-- font-awesome icons -->
        <link href="css/font-awesome.css" rel="stylesheet" type="text/css" media="all" />
        <!-- //font-awesome icons -->
        <!-- js -->
        <script src="js/jquery-1.11.1.min.js"></script>
        <!-- //js -->
        <link
        href='//fonts.googleapis.com/css?family=Ubuntu:400,300,300italic,400italic,500,500italic,700,
        700italic' rel='stylesheet' type='text/css'>
        <link
```

```
href='//fonts.googleapis.com/css?family=Open+Sans:400,300,300italic,400italic,600,600italic,
700,700italic,800,800italic' rel='stylesheet' type='text/css'>
<!-- start-smoth-scrolling -->


<script type="text/javascript" src="js/move-top.js"></script>
<script type="text/javascript" src="js/easing.js"></script>
<script type="text/javascript">
    jQuery(document).ready(function($) {
              $(".scroll").click(function(event){
                        event.preventDefault();
                        $('html,body').animate({scrollTop:$(this.hash).offset().top},1000);
              });
    });
</script>
<!-- start-back button disable -->

</head>

<body>
<!-- header -->
    <div class="agileits_header">

    <div class="product_list_header">

    </div>
<!-- script-for sticky-nav -->
    <script>
    $(document).ready(function() {
              var navoffeset=$(".agileits_header").offset().top;
              $(window).scroll(function(){
                        var scrollpos=$(window).scrollTop();
                        if(scrollpos >=navoffeset){
                                  $(".agileits_header").addClass("fixed");
                        }else{
                                  $(".agileits_header").removeClass("fixed");
                        }
              });

    });
    </script>



    <div class="w3l_banner_nav_right">


<div class="resp-tabs-container hor_1 checkout-right privacy about">
            <center><label class="control-label" style="color:green">Enter  the valid OTP
Sent To Your Mail ID</label></center></br>
          <form action="payment_action.php" method="POST" class="creditly-card-form
agileinfo_form">

    <section class="creditly-wrapper wthree, w3_agileits_wrapper">

    <div class="credit-card-wrapper">

            <div class="first-row form-group">



                    <div class="w3_agileits_card_number_grids">



                            <div class="w3_agileits_card_number_grid_right">

                                    <div class="controls">

                                            <label class="control-
label">OTP</label>

                                            <input


class="number credit-card-number form-control" maxlength="6" min="3" title="Enter Valid
OTP"


type="password" name="security_otp" id="security_otp"


placeholder="Enter OTP" required>

    <button class="submit" name="submit_otp" id="submit_otp"><span>Submit OTP and
```

```
confirm Order </span></button>
<?php




if(isset($_POST['submit_otp']))
{

    $otp_tped=$_POST['security_otp'];


    $security_code=$_SESSION['security_code'];

                        $customer_bank_info=mysqli_query($con,"SELECT * FROM
`tbl_bank_info` WHERE  status='VALID' and user_type_id=3");

                        $row_customer=mysqli_fetch_array($customer_bank_info);

                                  $temp_otp=$row_customer['temp_otp'];

                        if($otp_tped!=$temp_otp)
                        {



                                                      echo "<script>

                                                      alert('Wrong OTP .Please enter correct
OTP');



                                                      </script>";
                                  }
                                  else
                                  {


                                  $old_balance=$row_customer['balance_amount'];
```

```
                                                if($old_balance<$cart_tl)
                                                {
                                                echo "<script>

                                                        alert('Insufficient Balance in your
card');

                                                        window.location='payment.php';

                                                        </script>";
                                                }
                                                else
                                                {

                                                $new_balance=$old_balance-$cart_tl;

                                                $sql2=mysqli_query($con,"  UPDATE
`tbl_bank_info` SET `balance_amount`=$new_balance WHERE  CVV='$security_code' and
user_type_id=3 ");

                                                        if($sql2)
                                                                {




    $seller_bank_info=mysqli_query($con,"SELECT * FROM `tbl_bank_info` WHERE
status='VALID' and user_type_id=2 ");



    $row_seller=mysqli_fetch_array($seller_bank_info);

    $seller_old_bal=$row_seller['balance_amount'];

    $seller_new_bal=$seller_old_bal+$cart_tl;

                                                                $sql3=mysqli_query($con,"
UPDATE `tbl_bank_info` SET `balance_amount`=$seller_new_bal WHERE   user_type_id=2
");
```

```
                                                            if($sql3)
                                                            {


   $d_cutomer_cart=mysqli_query($con,"SELECT * FROM `tbl_customer_cart` WHERE
register_email='$mail' and status='VALID'");

   while($rowcart=mysqli_fetch_array($d_cutomer_cart))
                                                                   {

   $unique_ref_length = 6;

   $unique_ref_found = false;

   $possible_chars = "123456789";

                                              $possible_letters =
"ABCDEFGHIJKLMNOPQRSTUVWXYZ";

                                              // Until we find a unique reference,
keep generating new ones

                                              while (!$unique_ref_found) {

                                                     // Start with a blank
reference number

                                                     $unique_ref = "";

                                                     $unique_ref_c = "";

                                                     // Set up a counter to keep
track of how many characters have
```

```
                                                                // currently been added

                                                                $i = 0;



                                                                // Add random characters
from $possible_chars to $unique_ref

                                                                // until $unique_ref_length
is reached

                                                                          while ($i <
$unique_ref_length)

                                                                          {



                                                                                    // Pick
a random character from the $possible_chars list

                                                                                    $char =
substr($possible_chars, mt_rand(0, strlen($possible_chars)-1), 1);




    $unique_ref .= $char;


              $i++;

    }
    $j=0;
while ($j < 4)
{

    // Pick a random character from the $possible_chars list
    $char1 = substr($possible_letters, mt_rand(0, strlen($possible_letters)-1), 1);

                                                      $unique_ref_c .= $char1;

    $j++;
}

$customer_ref_id=$unique_ref."-".$unique_ref_c;


                                        $query = "SELECT `order_ref_no` FROM
`tbl_customer_order`


WHERE `order_ref_no`='".$customer_ref_id."'";

                                                        $result =
mysqli_query($con,$query);

                                                                if
(mysqli_num_rows($result)==0)

                                                                {



                                                                        //
We've found a unique number. Lets set the $unique_ref_found

                                                                        //
variable to true and exit the while loop


$unique_ref_found = true;
}}
                                                //$customer_ref_id=$unique_ref."-
".$unique_ref_c;

$stock_product_id=$rowcart['stock_product_id'];

$purchase_qty=$rowcart['quantity'];

$purchase_price=$rowcart['total_price'];

$customize_comments=$rowcart['cart_custmize_comment'];

$sql_order_prod=mysqli_query($con,"  INSERT INTO `tbl_customer_order` (
`register_email`, `stock_product_id`, `delv_adres_id`, `purchase_qty`, `purchase_price`,
`order_date`, `delivery_date`, `customize_comments`, `status`,`order_ref_no`) VALUES
('$mail', $stock_product_id, $delvr_address_id, $purchase_qty, $purchase_price, now(), '0000-
00-00', '$customize_comments', 'NEW','$customer_ref_id') ");
```

```
                        if($sql_order_prod)

                                {

    $sql_del_cart=mysqli_query($con," DELETE FROM `tbl_customer_cart` WHERE
register_email='$mail' and stock_product_id=$stock_product_id and status='VALID'");

                                                            if($sql_del_cart)

                                            {

    $seller_stock_view=mysqli_query($con,"SELECT * FROM `tbl_seller_stock` WHERE
status='VALID' and stock_product_id=$stock_product_id ");


    $rowstock_count=mysqli_fetch_array($seller_stock_view);


    $old_stock_count=$rowstock_count['stock_item_count'];

                                            $new_stock_count=$old_stock_count-
$purchase_qty;




    $seller_stock_update=mysqli_query($con,"UPDATE `tbl_seller_stock` SET
`stock_item_count` = $new_stock_count WHERE `tbl_seller_stock`.`stock_product_id` =
$stock_product_id ");




                                            //UPDATE `tbl_seller_stock` SET
`stock_item_count` = '2' WHERE `tbl_seller_stock`.`stock_product_id` = 20;

                                                    if($seller_stock_update)
                        {

                                                    echo "<script>


    alert('Order placed successfully');
```

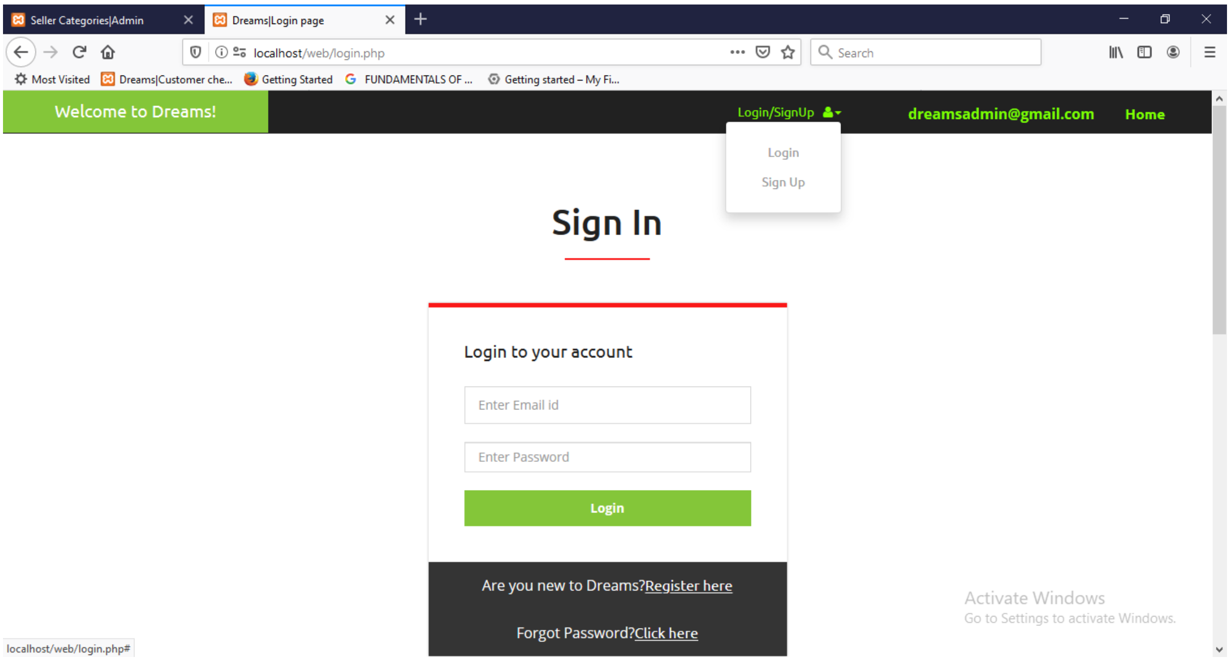
```
	window.location='customer_view_orders.php';


	</script>";
			}
			}}else{	echo "<script>


	alert('failed');
	}
}//end while

}//end if sql3

else
{
			echo "<script>

				alert('Transaction failed');

				window.location='cart_view.php';

				</script>";



									}


								}//end if sql2


						}//end else balance check

					}//end else otp match
				}//isset submit otp button

?>

<!-- Bootstrap Core JavaScript -->
<script src="js/bootstrap.min.js"></script>
	<script type="text/javascript">
    $(document).ready(function() {
      //Horizontal Tab
```

```
        $('#parentHorizontalTab').easyResponsiveTabs({
            type: 'default', //Types: default, vertical, accordion
            width: 'auto', //auto or any width like 600px
            fit: true, // 100% fit in a container
            tabidentify: 'hor_1', // The tab groups identifier
            activate: function(event) { // Callback function if tab is switched
                var $tab = $(this);
                var $info = $('#nested-tabInfo');
                var $name = $('span', $info);
                $name.text($tab.text());
                $info.show();
            }
        });
    });
</script>
<!-- credit-card -->
                        <script type="text/javascript" src="js/creditly.js"></script>
        <link rel="stylesheet" href="css/creditly.css" type="text/css" media="all" />

                        <script type="text/javascript">
                                        $(function() {
                                          var creditly = Creditly.initialize(
                                                          '.creditly-wrapper .expiration-month-and-year',
                                                          '.creditly-wrapper .credit-card-number',
                                                          '.creditly-wrapper .security-code',
                                                          '.creditly-wrapper .card-type');

                                          $(".creditly-card-form .submit").click(function(e) {
                                                        e.preventDefault();
                                                        var output = creditly.validate();
                                                        if (output) {
                                                          // Your validated credit card output
                                                          console.log(output);
                                                        }
                                          });
                                        });
                        </script>
       <!-- //credit-card -->

<!-- //js -->
```

```
<!-- script-for sticky-nav -->
      <script>
      $(document).ready(function() {
                       var navoffeset=$(".agileits_header").offset().top;
                       $(window).scroll(function(){
                                     var scrollpos=$(window).scrollTop();
                                     if(scrollpos >=navoffeset){
                                                    $(".agileits_header").addClass("fixed");
                                     }else{
                                                    $(".agileits_header").removeClass("fixed");
                                     }
                       });

      });
      </script>
<!-- //script-for sticky-nav -->
<!-- start-smoth-scrolling -->
<script type="text/javascript" src="js/move-top.js"></script>
<script type="text/javascript" src="js/easing.js"></script>
<script type="text/javascript">
      jQuery(document).ready(function($) {
                      $(".scroll").click(function(event){
                                     event.preventDefault();
                                     $('html,body').animate({scrollTop:$(this.hash).offset().top},1000);
                      });
      });
</script>
<!-- start-smoth-scrolling -->
<!-- Bootstrap Core JavaScript -->
<script src="js/bootstrap.min.js"></script>
<script>
$(document).ready(function(){
    $(".dropdown").hover(
        function() {
            $('.dropdown-menu', this).stop( true, true ).slideDown("fast");
            $(this).toggleClass('open');
        },
        function() {
            $('.dropdown-menu', this).stop( true, true ).slideUp("fast");
            $(this).toggleClass('open');
        }
    );
});
</script>
<!-- here stars scrolling icon -->
      <script type="text/javascript">
                  $(document).ready(function() {
                              /*
                                          var defaults = {
                                          containerID: 'toTop', // fading element id
                                          containerHoverID: 'toTopHover', // fading element hover id
                                          scrollSpeed: 1200,
                                          easingType: 'linear'
                                          };
                              */

                              $().UItoTop({ easingType: 'easeOutQuart' });

                              });
      </script>
<!-- //here ends scrolling icon -->
<script src="js/minicart.js"></script>
<script>
                  paypal.minicart.render();

                  paypal.minicart.cart.on('checkout', function (evt) {
                              var items = this.items(),
                                          len = items.length,
                                          total = 0,
                                          i;
                              for (i = 0; i < len; i++) {
                                          total += items[i].get('quantity');
                              }
                              if (total < 3) {
                                          alert('The minimum order quantity is 3. Please add more to
your shopping cart before checking out');
                                          evt.preventDefault();
                              }
                  });
    </script>
</body>
</html>

<?php }

else{

header("location:login.php");

}

 ?>
```

SCREENSHOTS

Login Page



Sign In

Login to your account

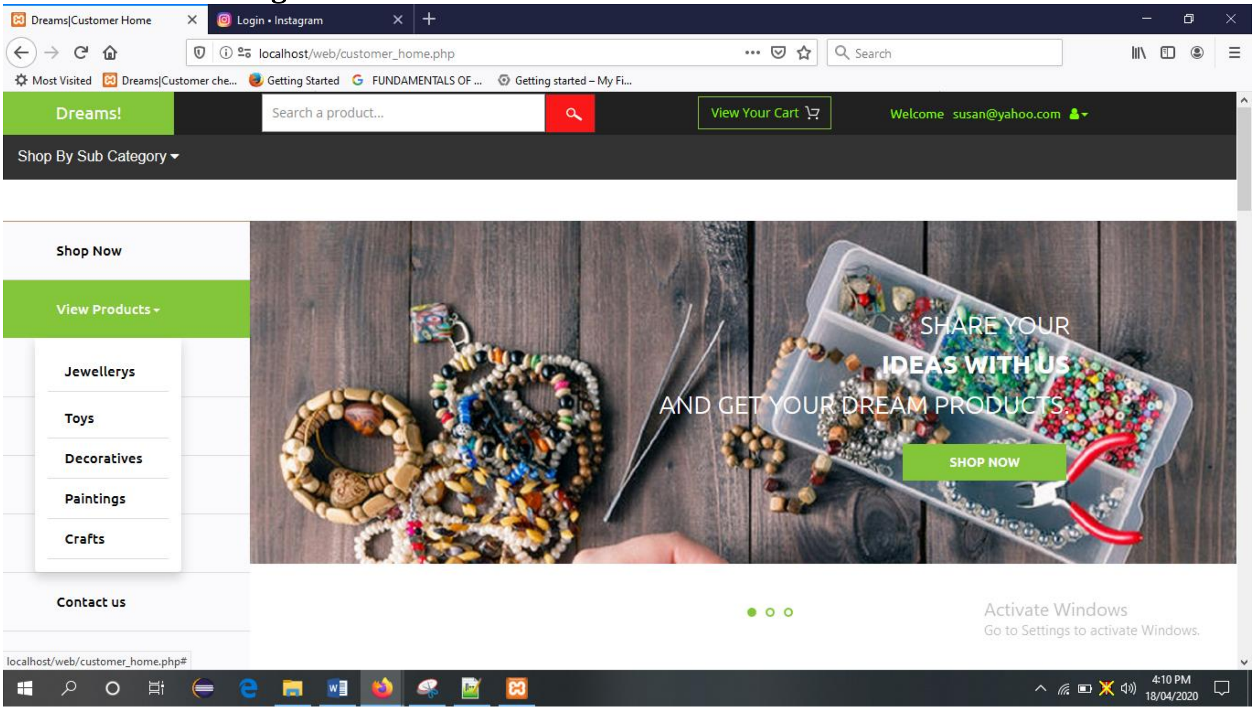
Enter Email id

Enter Password

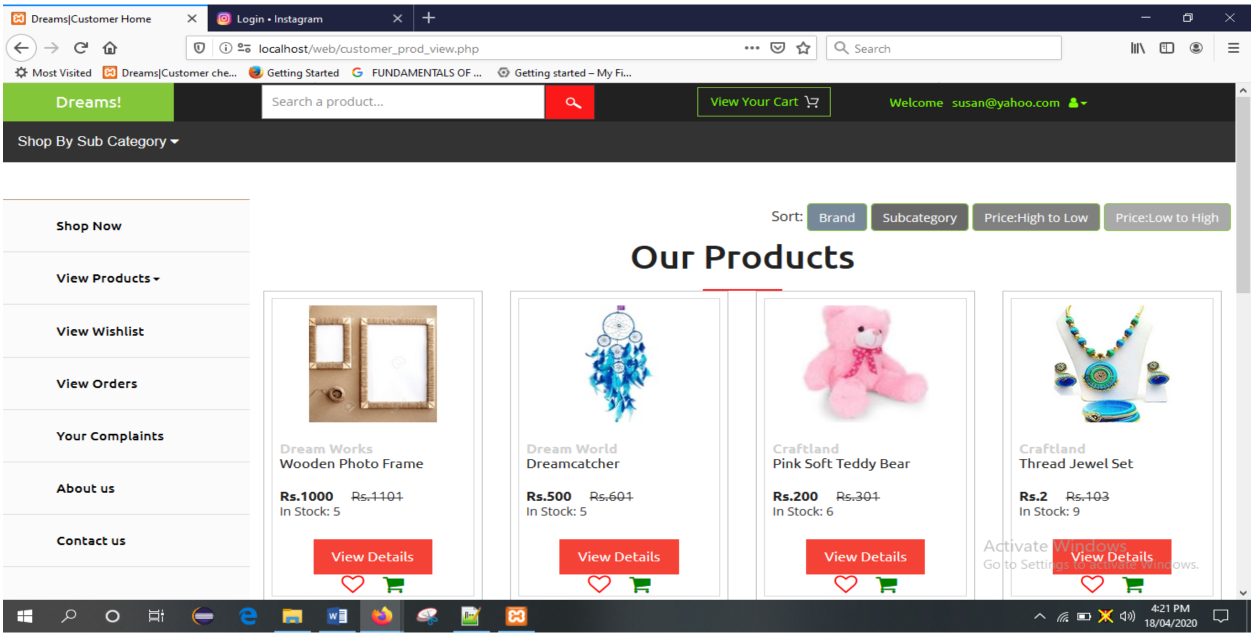
Are you new to Dreams?Register here

Forgot Password?Click here

Customer Home Page



Products view page



Our Products

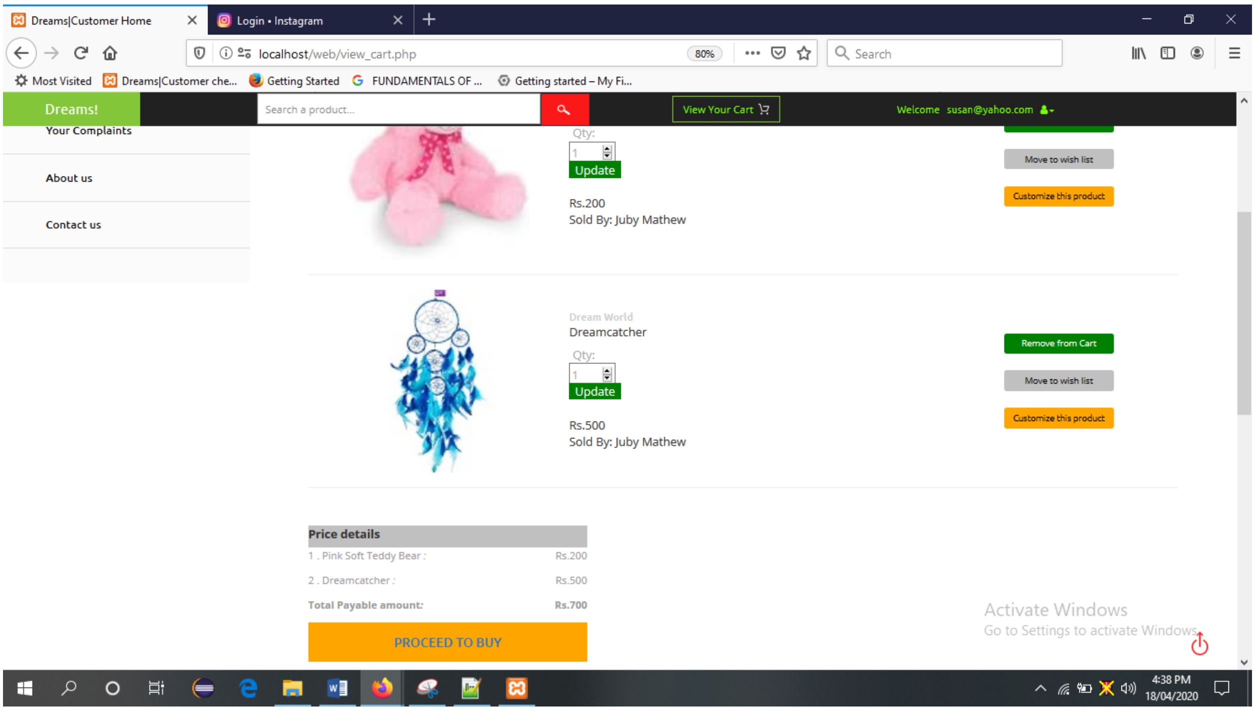
Dream Works
Wooden Photo Frame
Rs.1000 ~~Rs.1101~~
In Stock: 5

Dream World
Dreamcatcher
Rs.500 ~~Rs.601~~
In Stock: 5

Craftland
Pink Soft Teddy Bear
Rs.200 ~~Rs.301~~
In Stock: 6

Craftland
Thread Jewel Set
Rs.2 ~~Rs.103~~
In Stock: 9

Cart Page

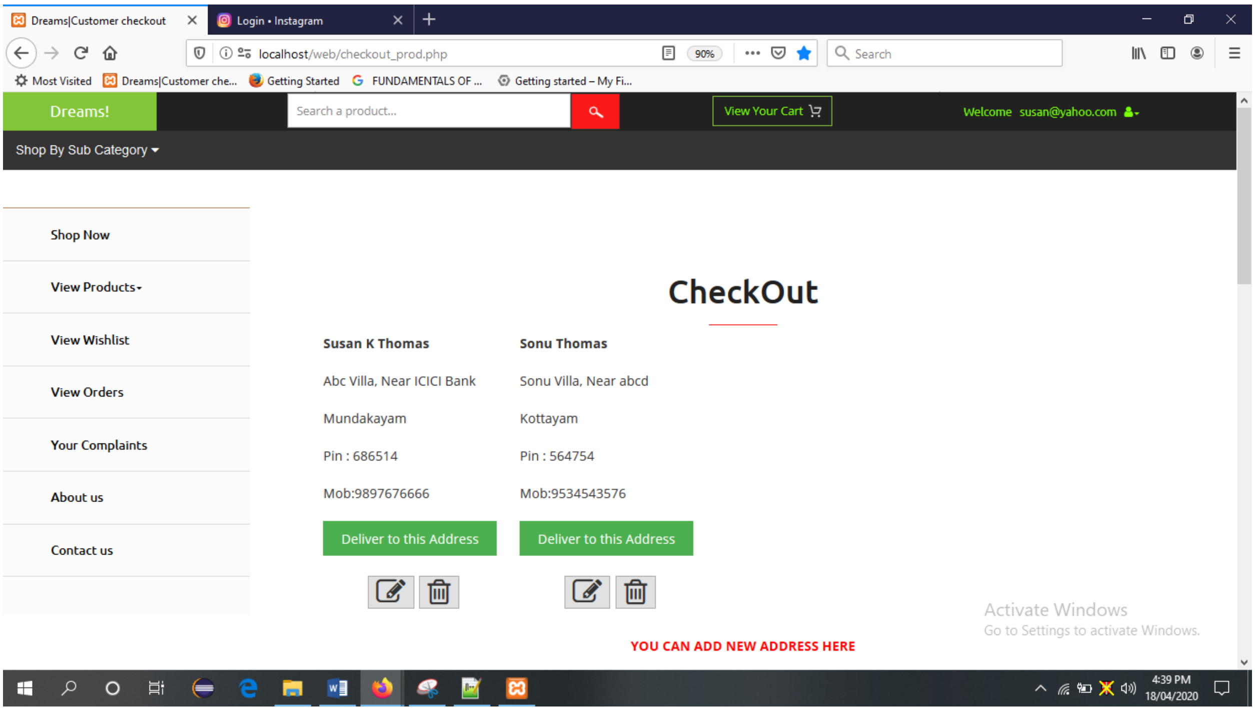


Qty: 1
Rs.200
Sold By: Juby Mathew

Dream World
Dreamcatcher
Qty: 1
Rs.500
Sold By: Juby Mathew

Price details	
1 . Pink Soft Teddy Bear :	Rs.200
2 . Dreamcatcher :	Rs.500
Total Payable amount:	Rs.700

Delivery address page

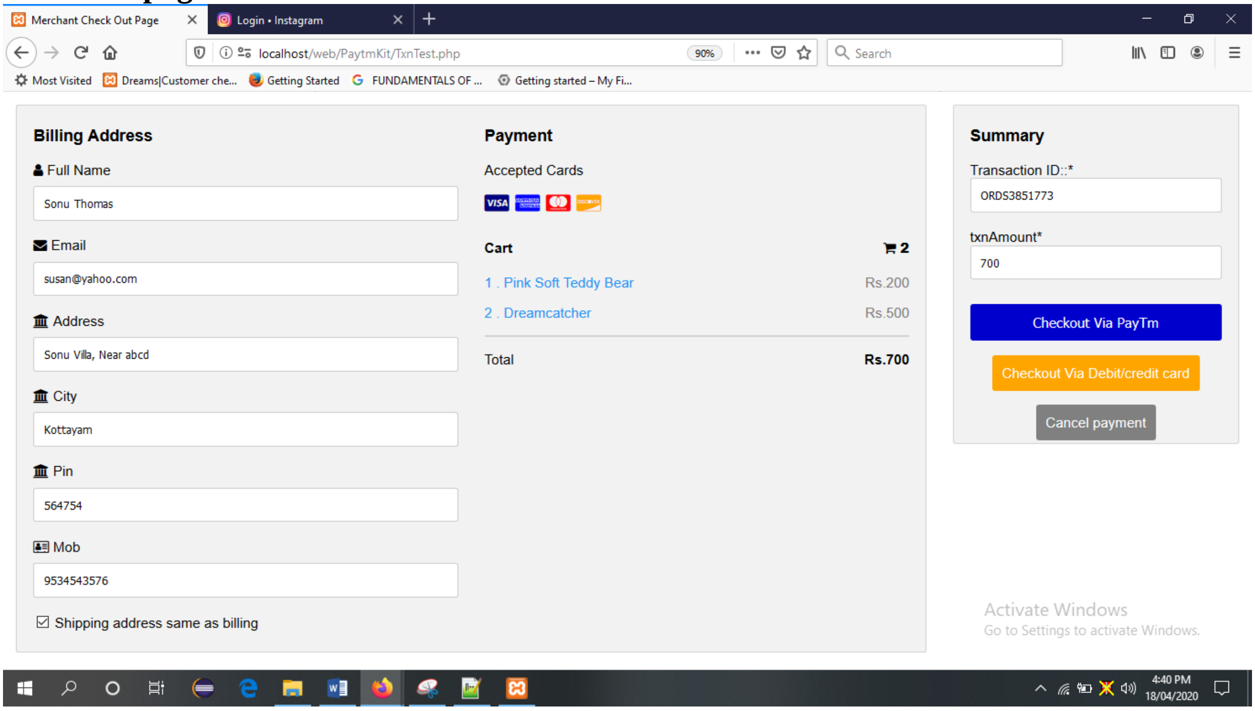


CheckOut

Susan K Thomas	Sonu Thomas
Abc Villa, Near ICICI Bank	Sonu Villa, Near abcd
Mundakayam	Kottayam
Pin : 686514	Pin : 564754
Mob:9897676666	Mob:9534543576
Deliver to this Address	Deliver to this Address

YOU CAN ADD NEW ADDRESS HERE

Checkout page



Billing Address

Full Name: Sonu Thomas

Email: susan@yahoo.com

Address: Sonu Villa, Near abcd

City: Kottayam

Pin: 564754

Mob: 9534543576

☑ Shipping address same as billing

Payment

Accepted Cards

Cart	2
1 . Pink Soft Teddy Bear	Rs.200
2 . Dreamcatcher	Rs.500
Total	Rs.700

Summary

Transaction ID::* ORDS3851773

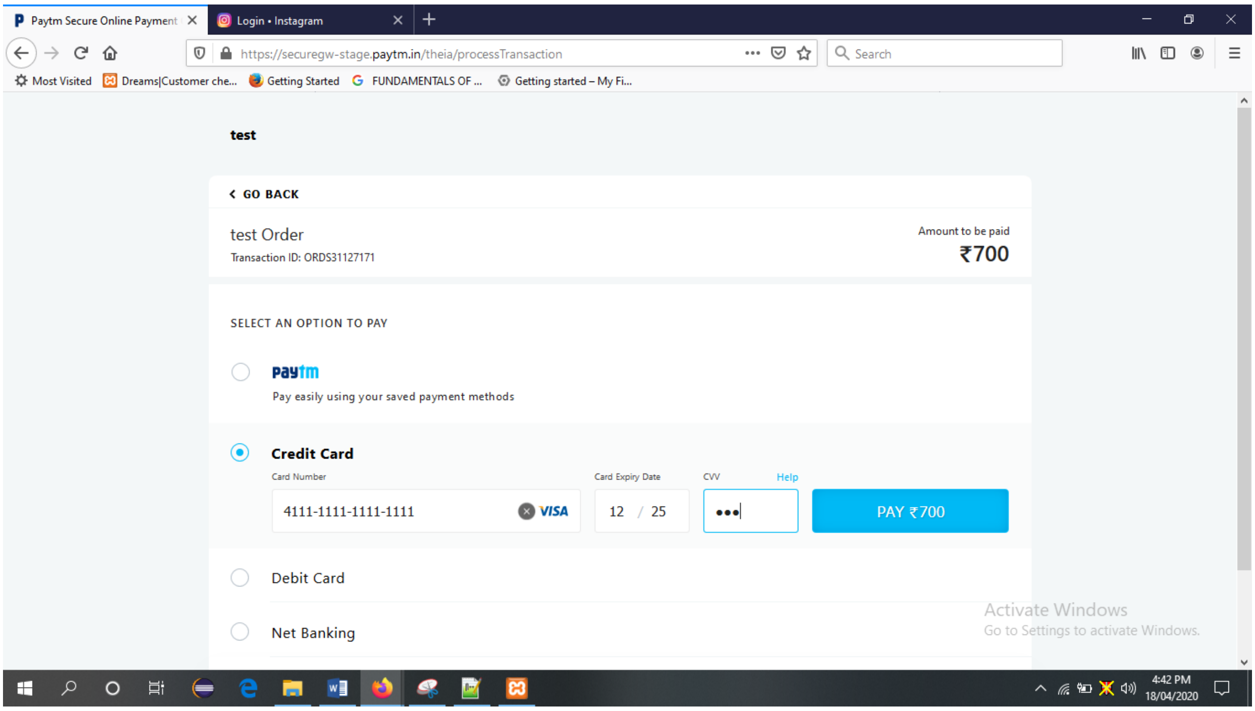
txnAmount* 700

Checkout Via PayTm

Checkout Via Debit/credit card

Cancel payment

Payment via Paytm gateway



Paytm Secure Online Payment
Login • Instagram
https://securegw-stage.paytm.in/theia/processTransaction
Search
Most Visited
Dreams|Customer che...
Getting Started
FUNDAMENTALS OF ...
Getting started – My Fi...

test

‹ GO BACK

test Order

Transaction ID: ORDS31127171

Amount to be paid

₹700

SELECT AN OPTION TO PAY

○ Paytm

Pay easily using your saved payment methods

● Credit Card

Card Number: 4111-1111-1111-1111 VISA

Card Expiry Date: 12 / 25

CVV: ••• Help

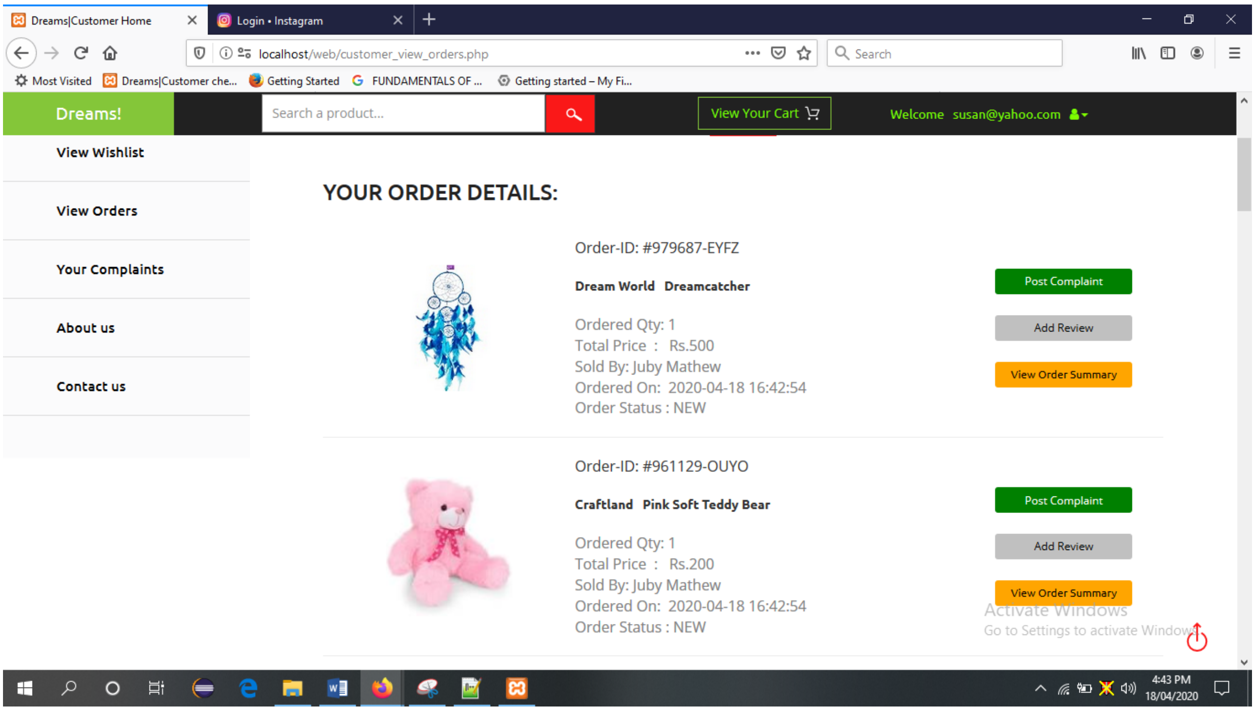
PAY ₹700

○ Debit Card

○ Net Banking

Activate Windows
Go to Settings to activate Windows.
4:42 PM
18/04/2020

Order Details Page



YOUR ORDER DETAILS:

Order-ID: #979687-EYFZ

Dream World Dreamcatcher

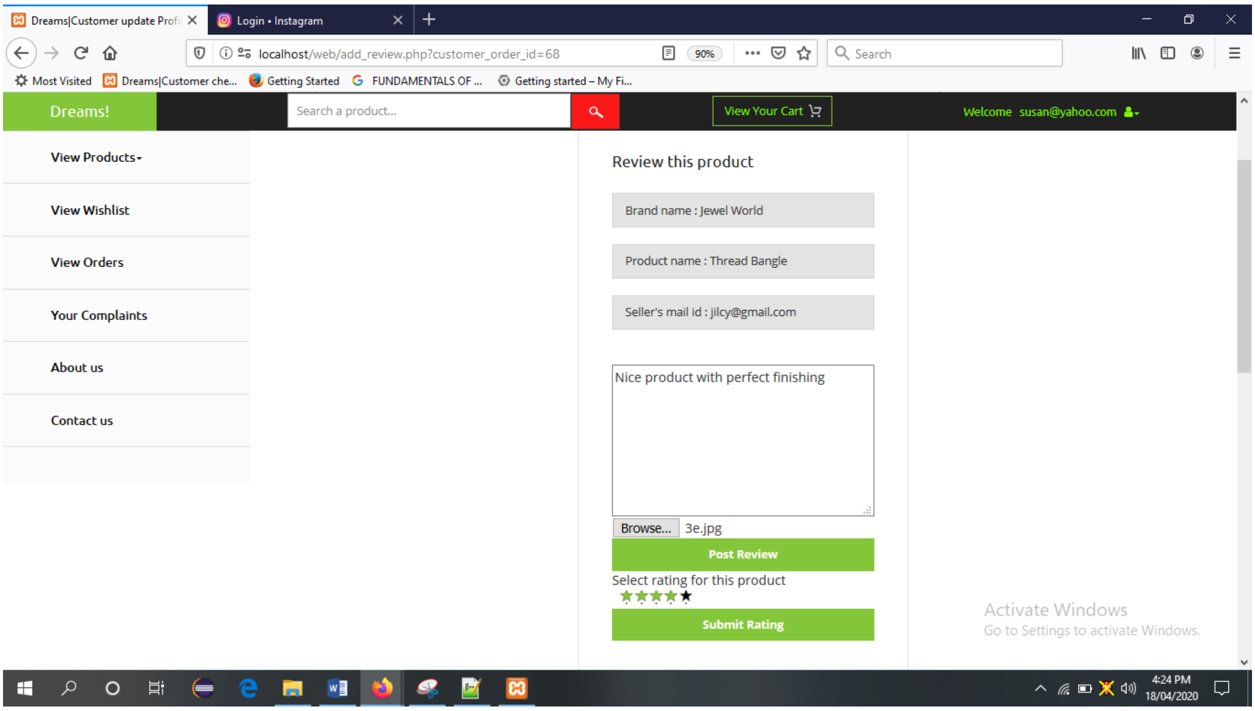
Ordered Qty: 1
Total Price : Rs.500
Sold By: Juby Mathew
Ordered On: 2020-04-18 16:42:54
Order Status : NEW

Order-ID: #961129-OUYO

Craftland Pink Soft Teddy Bear

Ordered Qty: 1
Total Price : Rs.200
Sold By: Juby Mathew
Ordered On: 2020-04-18 16:42:54
Order Status : NEW

Add Review and rating page



Review this product

Brand name : Jewel World

Product name : Thread Bangle

Seller's mail id : jilcy@gmail.com

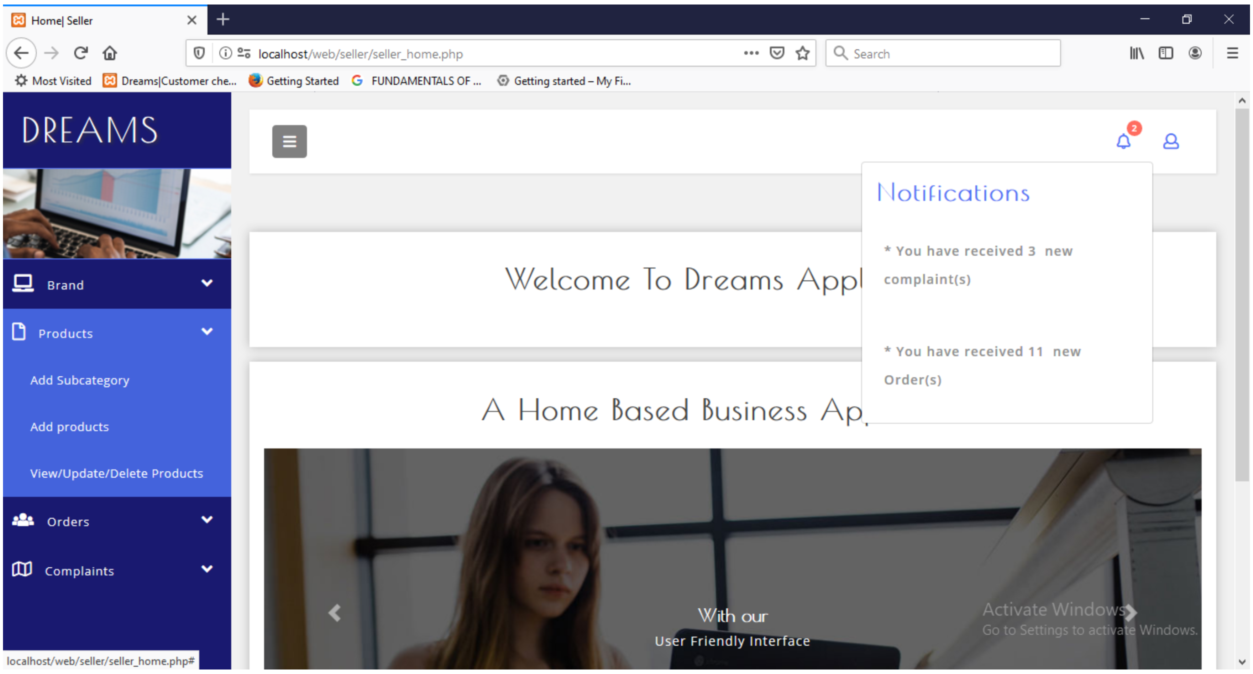
Nice product with perfect finishing

3e.jpg

Select rating for this product

★★★★★

Seller Home Page

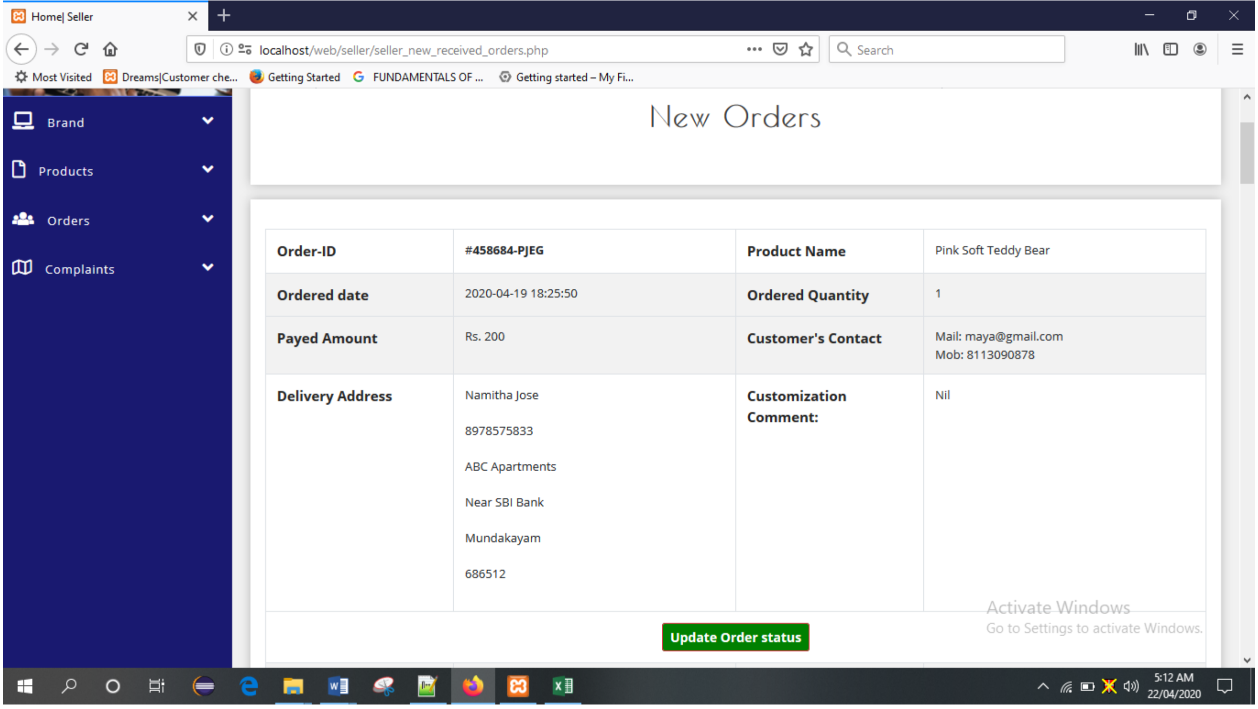


Notifications

* You have received 3 new complaint(s)

* You have received 11 new Order(s)

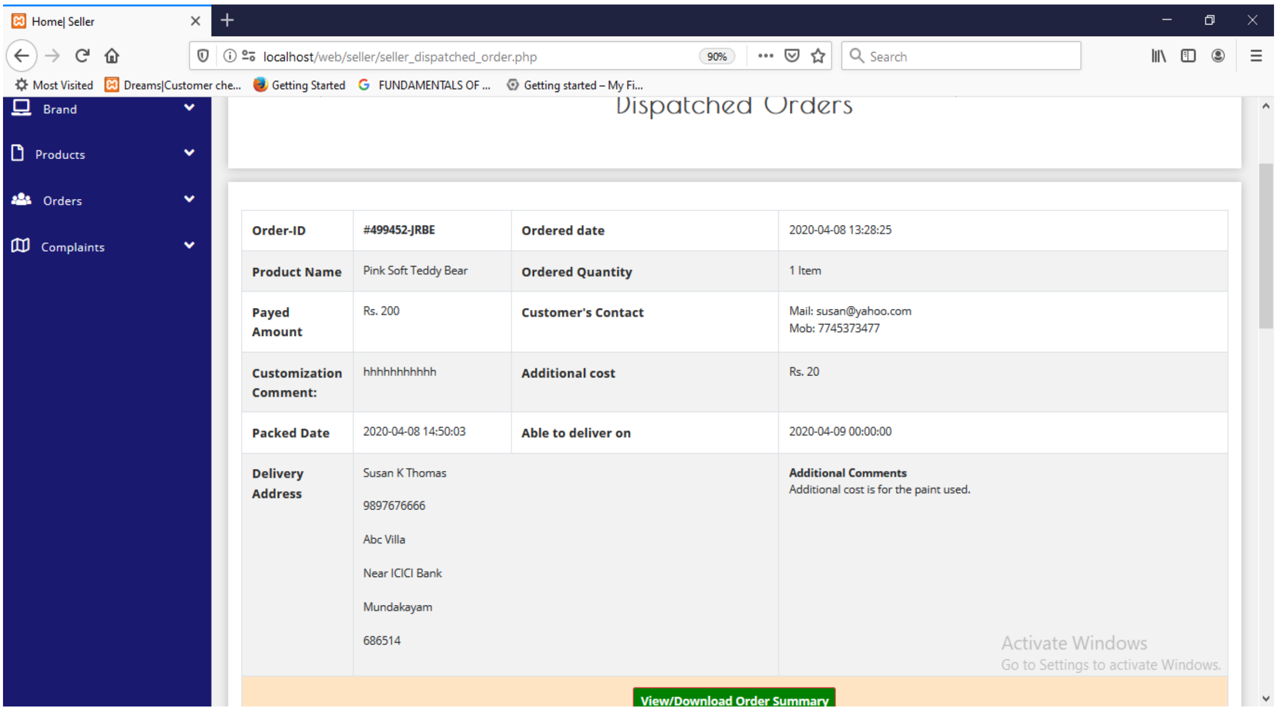
Newly received order's page



New Orders

Order-ID	**#458684-PJEG**	**Product Name**	Pink Soft Teddy Bear
Ordered date	2020-04-19 18:25:50	**Ordered Quantity**	1
Payed Amount	Rs. 200	**Customer's Contact**	Mail: maya@gmail.com Mob: 8113090878
Delivery Address	Namitha Jose 8978575833 ABC Apartments Near SBI Bank Mundakayam 686512	**Customization Comment:**	Nil

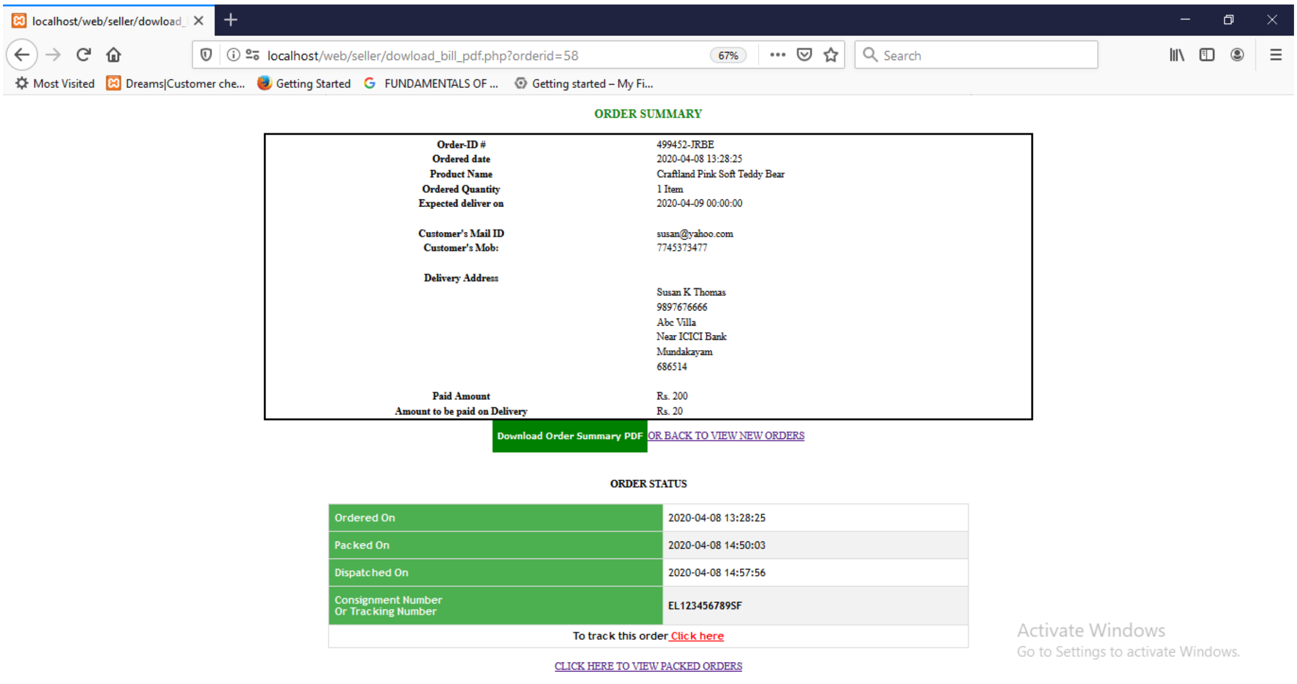
Dispatched order's page


Home| Seller
localhost/web/seller/seller_dispatched_order.php
90%
Search
Most Visited
Dreams|Customer che...
Getting Started
FUNDAMENTALS OF ...
Getting started – My Fi...
Brand
Products
Orders
Complaints
View/Download Order Summary
Activate Windows
Go to Settings to activate Windows.

Dispatched Orders

Order-ID	**#499452-JRBE**	**Ordered date**	2020-04-08 13:28:25
Product Name	Pink Soft Teddy Bear	**Ordered Quantity**	1 Item
Payed Amount	Rs. 200	**Customer's Contact**	Mail: susan@yahoo.com Mob: 7745373477
Customization Comment:	hhhhhhhhhh	**Additional cost**	Rs. 20
Packed Date	2020-04-08 14:50:03	**Able to deliver on**	2020-04-09 00:00:00
Delivery Address	Susan K Thomas 9897676666 Abc Villa Near ICICI Bank Mundakayam 686514		**Additional Comments** Additional cost is for the paint used.

Order Summary with tracking details page



localhost/web/seller/dowload_
localhost/web/seller/dowload_bill_pdf.php?orderid=58
67%
Search
Most Visited
Dreams|Customer che...
Getting Started
FUNDAMENTALS OF ...
Getting started – My Fi...

ORDER SUMMARY

Order-ID #	499452-JRBE
Ordered date	2020-04-08 13:28:25
Product Name	Craftland Pink Soft Teddy Bear
Ordered Quantity	1 Item
Expected deliver on	2020-04-09 00:00:00
Customer's Mail ID	susan@yahoo.com
Customer's Mob:	7745373477
Delivery Address	Susan K Thomas 9897676666 Abc Villa Near ICICI Bank Mundakayam 686514
Paid Amount	Rs. 200
Amount to be paid on Delivery	Rs. 20

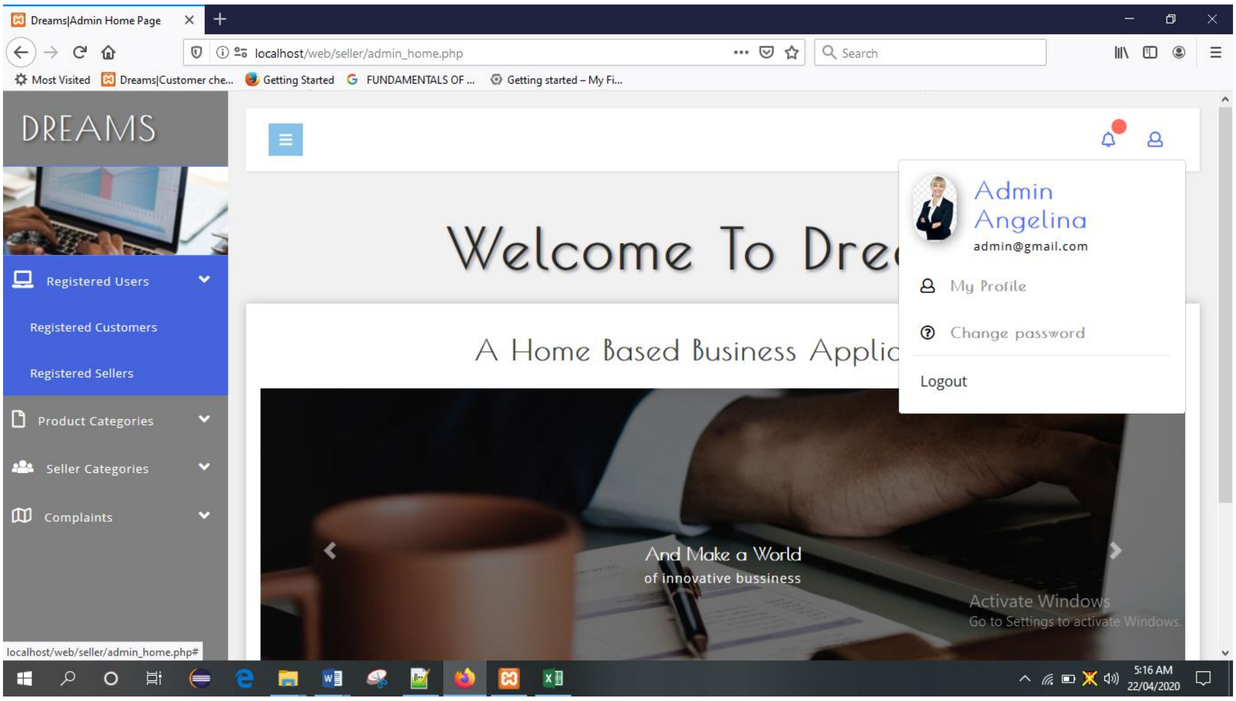
ORDER STATUS

Ordered On	2020-04-08 13:28:25
Packed On	2020-04-08 14:50:03
Dispatched On	2020-04-08 14:57:56
Consignment Number Or Tracking Number	EL123456789SF

To track this order Click here

CLICK HERE TO VIEW PACKED ORDERS
Activate Windows
Go to Settings to activate Windows.

Admin Home Page

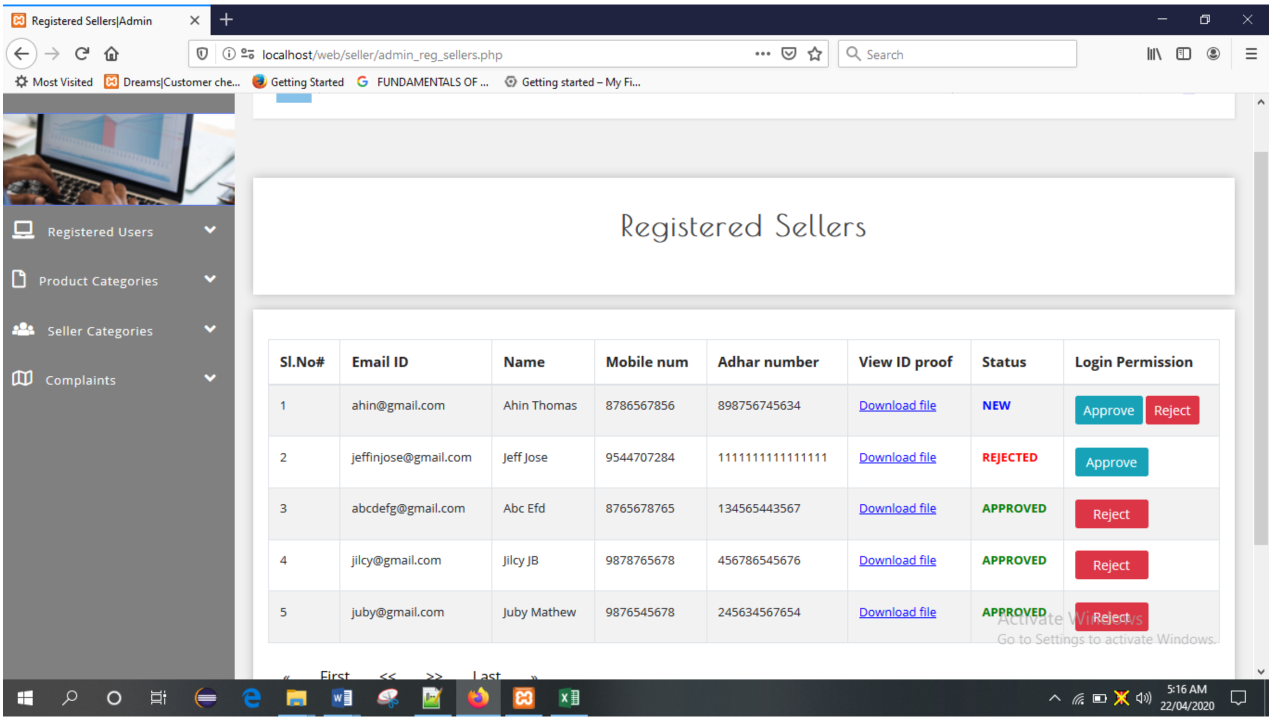


Welcome To Dre

A Home Based Business Applic

And Make a World
of innovative bussiness

Registered seller's page

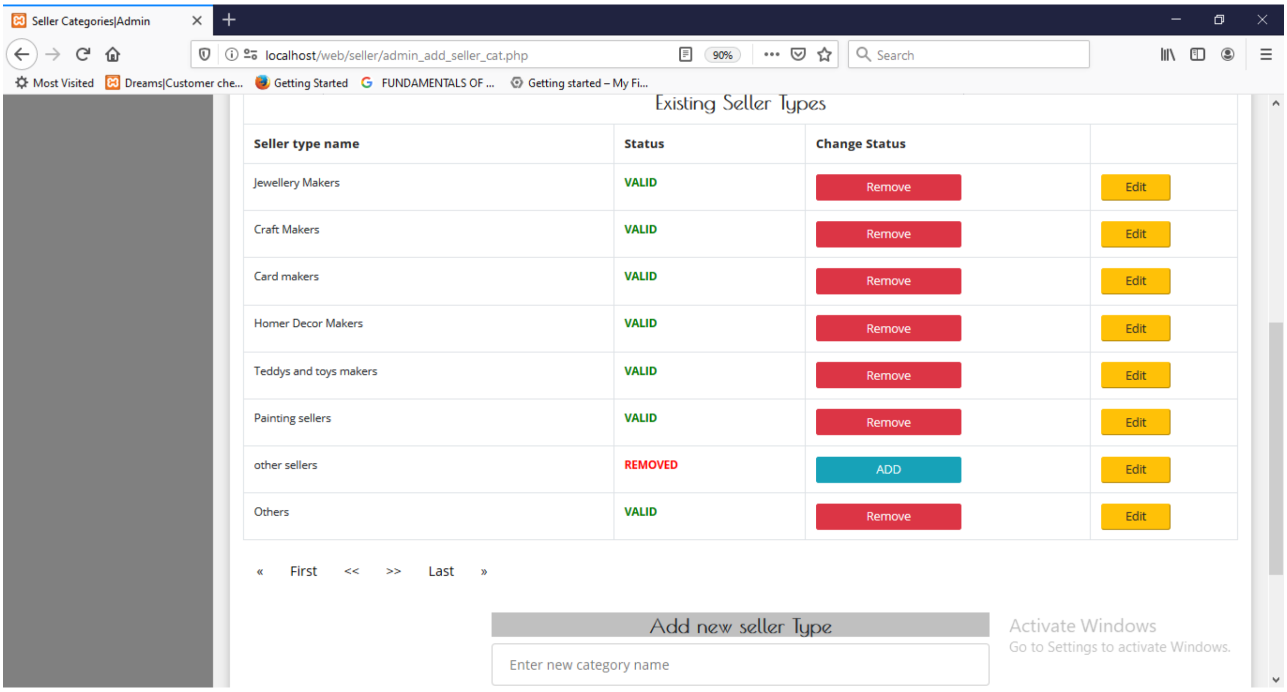


Registered Sellers

Sl.No#	Email ID	Name	Mobile num	Adhar number	View ID proof	Status	Login Permission
1	ahin@gmail.com	Ahin Thomas	8786567856	898756745634	Download file	NEW	Approve Reject
2	jeffinjose@gmail.com	Jeff Jose	9544707284	1111111111111111	Download file	REJECTED	Approve
3	abcdefg@gmail.com	Abc Efd	8765678765	134565443567	Download file	APPROVED	Reject
4	jilcy@gmail.com	Jilcy JB	9878765678	456786545676	Download file	APPROVED	Reject
5	juby@gmail.com	Juby Mathew	9876545678	245634567654	Download file	APPROVED	Reject

« First << >> Last »

View/add/update seller types page



Existing Seller Types

Seller type name	Status	Change Status	
Jewellery Makers	VALID	Remove	Edit
Craft Makers	VALID	Remove	Edit
Card makers	VALID	Remove	Edit
Homer Decor Makers	VALID	Remove	Edit
Teddys and toys makers	VALID	Remove	Edit
Painting sellers	VALID	Remove	Edit
other sellers	REMOVED	ADD	Edit
Others	VALID	Remove	Edit

« First << >> Last »

Add new seller Type

Enter new category name