LaraComm Version 0.1

A project submitted to the

ज्ञानगंगा घरोघरी

School of Computer Science of

Yashwantrao Chavan Maharashtra Open University,

Nashik

In Partial Fulfillment of

The Requirements for the Degree of

Bachelor of Computer Application (B.C.A)

By

Mehul Bawadia

2014 - 2015

Certificate of Completion

This is to certify that I, ***Mr. Mehul Bawadia***, have completed the project on ***LaraComm Version 0.1*** as a partial fulfillment of the ***Graduate Degree of Bachelors in Computer Application (B. C. A)***

Signature:

Project Guide: Nirav Chheda

St. Angelo's Computer Education

Date:

Seal:

Internal Examiner	External Examiner:
Signature:	Signature:
Name:	Name:
Date:	Date:

Acknowledgment

I take this opportunity to thank all those who have been directly or indirectly related to this project.

A number of people with their innovative ideas have contributed to this Project in more than one way and we owe them all a sincere acknowledgements.

Special thanks to Nirav Chheda Sir, my Internal Guide, for his unending co-operation and encouragement throughout the project span, for his unending support and guidance in every responsibility I accepted.

I also would like to thank people at StackExchange Community who were always there to help me whenever I was in need.

And last but not the least I would like to thank all our friends for their friendly criticism and honest suggestions. I acknowledge this constant support and encouragement and I am thankful to all of them for their valuable contribution to this project.

Sincerely express my gratitude.

Synopsis

LaraComm Version 0.1 is an E-Commerce web application for all kinds of jewelry products build on the PHP Framework called Laravel created by Mr. Taylor Otwell.

There are two entities involved in the whole application, viz. Administrator and the Customer or the end-user.

The administrator is the one who monitors and maintains the whole application

Customer or the end user is the one who accesses the application on the internet.

INDEX

Chapter 1. Introduction

The following points will be covered in this chapter.

1. Introduction of E-Commerce
2. Brief history of E-Commerce
3. Brief introduction about LaraComm Version 0.1
4. Brief Introduction about MVC Framework

Introduction to E-Commerce

What is E-Commerce?

E-commerce, short for **electronic commerce**, is trading in products or services using computer networks, such as the Internet. Electronic commerce draws on technologies such as mobile commerce, electronic funds transfer, supply chain management, Internet marketing, online transaction processing, electronic data interchange (EDI), inventory management systems, and automated data collection systems. Modern electronic commerce typically uses the World Wide Web for at least one part of the transaction's life cycle, although it may also use other technologies such as e-mail.

In **India**, the Information Technology Act 2000 governs the basic applicability of e-commerce. It is based upon UNCITRAL Model but is not a comprehensive legislation to deal with e-commerce related activities in India. Further, e-commerce laws and regulations in India [42] are also supplemented by different laws of India as applicable to the field of e-commerce. For instance, e-commerce relating to pharmaceuticals, healthcare, traveling, etc. are governed by different laws though the information technology act, 2000 prescribes some common requirements for all these fields. The competition commission of India (CCI) regulates anti competition and anti-trade practices in e-commerce fields in India.[43] Some stakeholders have decided to approach courts and CCI against e-commerce websites to file complaint about unfair trade practices and predatory pricing by such e-commerce websites.

Brief history of E-Commerce?

The history of e-commerce has continued up to the present day but is generally considered to also predate the period in which the Internet was available. Before personal computers had been introduced which could be used practically and relatively inexpensively by a wide range of consumers for Internet commerce, the history of e-commerce turned on technology which was used between different businesses which had the funds to use them, rather than between consumers and businesses.

The history of e-commerce includes the aspect of OS commerce, or open source commerce, as can be used freely by an array of people. In this regard, the late 1970s saw the development, prior to anything resembling Internet commerce, of methods for businesses to process their transactions with each other using electronic means.

Strictly speaking, a range of the technologically-driven innovations of the 1980s, such as ATMs, can also be considered as parts of the history of e-commerce, albeit one without the same access to computing on the part of consumers as was enjoyed by the large businesses serving them.

The modern phase of e-commerce was marked by a shift to a more open source, Operating System commerce approach, as began with the 1990 creation of a web browser for searching the World Wide Web, though Internet commerce as such only became allowable in 1991. Moreover, OS commerce of this kind only became somewhat feasible with the Internet’s burst of popularity in 1994, and Internet commerce had become fairly widespread as a business model as of 2000.

What is LaraComm?

LaraComm is an E-Commerce web application build on PHP MVC Framework called Laravel which is built by Mr. Taylor Otwell. More details on MVC is on the next page.

LaraComm has 2 entities attached with it viz. Administrator and the Customer.

Administrator manages, monitors and controls the whole application.

Customer is the end user who is accessing the whole application searching the product to meet his needs.

The administrator can add / edit / delete anything he wants by going to the admin panel.

The LaraComm is built on Twitter Bootstrap Framework, meaning it can easily fit into your mobile devices without any much of hassles.

What is MVC?

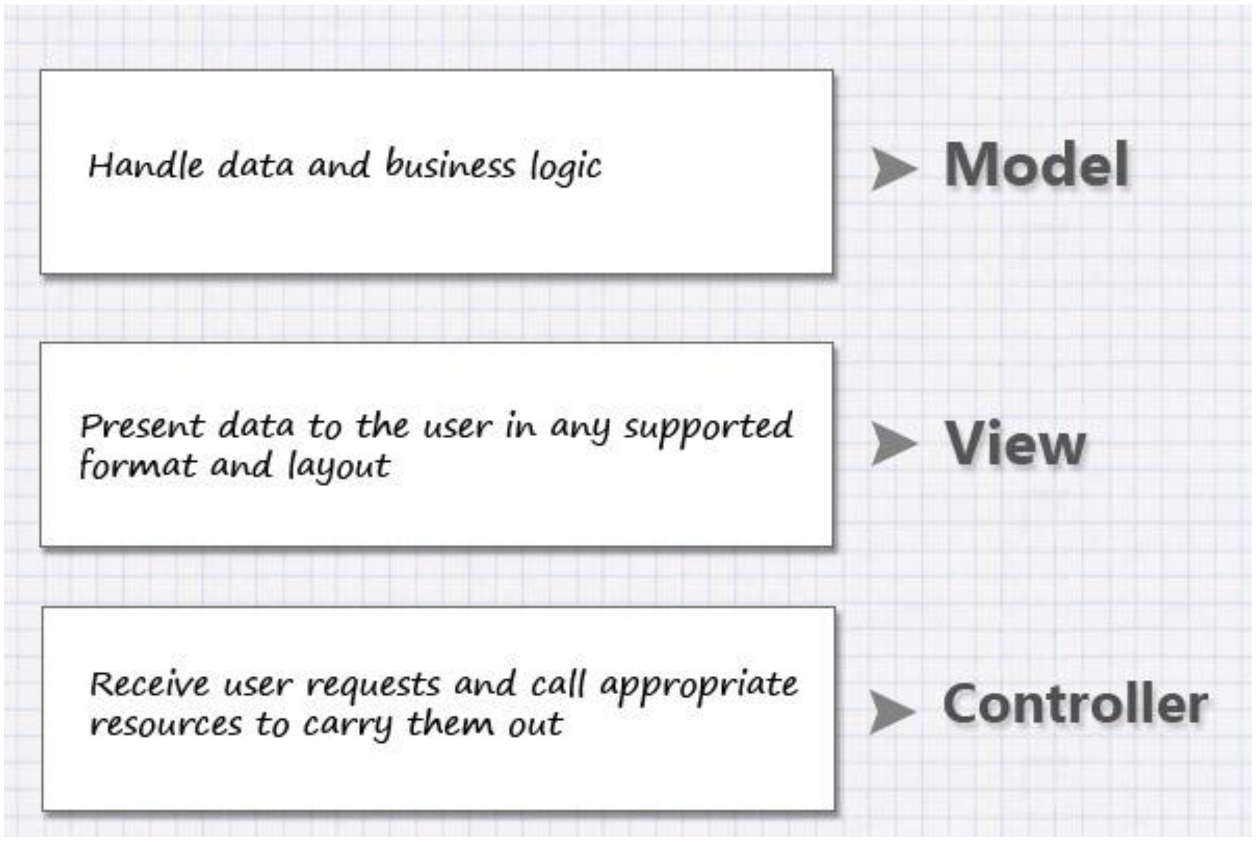
Model-View-Controller (MVC) is probably one of the most quoted patterns in the web programming world in recent years. Anyone currently working in anything related to web application development will have heard or read the acronym hundreds of times. Today, we'll clarify what MVC means, and why it has become so popular.

It was first described in 1979 and, obviously, the context was a little bit different. The concept of web application did not exist. Tim Berners Lee sowed the seeds of World Wide Web in the early nineties and changed the world forever. The pattern we use today for web development is an adaptation of the original pattern.

The wild popularization of this structure for web applications is due to its inclusion in two development frameworks that have become immensely popular: Struts and Ruby on Rails. These two environments marked the way for the hundreds of frameworks created later.

MVC for Web Applications

The idea behind the Model-View-Controller architectural pattern is simple: we must have the following responsibilities clearly separated in our application:


Handle data and business logic
Model
Present data to the user in any supported format and layout
View
Receive user requests and call appropriate resources to carry them out
Controller

The application is divided into these three main components, each one in charge of different tasks.

Controller

The **Controller** manages the user requests (received as HTTP GET or POST requests when the user clicks on GUI elements to perform actions). Its main function is to call and coordinate the necessary resources/objects needed to perform the user action. Usually the controller will call the appropriate model for the task and then selects the proper view.

- A controller offers facilities to change the state of the model. The controller interprets the mouse and keyboard inputs from the user, commanding the model and/or the view to change as appropriate.

- A controller is the means by which the user interacts with the application. A controller accepts input from the user and instructs the model and view to perform actions based on that input. In effect, the controller is responsible for mapping end-user action to application response.

- The controller translates interactions with the view into actions to be performed by the model. In a stand-alone GUI client, user interactions could be button clicks or menu selections, whereas in a Web application they appear as HTTP GET and POST requests. The actions performed by the model include activating business processes or changing the state of the model. Based on the user interactions and the outcome of the model actions, the controller responds by selecting an appropriate view.

- The controller is the piece that manages user interaction with the model. It provides the mechanism by which changes are made to the state of the model.

Model

The **Model** is the data and the rules applying to that data, which represent concepts that the application manages. In any software system, everything is modeled as data that we handle in a certain way. What is a user, a message or a book for an application? Only data that must be handled according to specific rules (date cannot be in the future, e-mail must have a specific format, name cannot be more than x characters long, etc.).

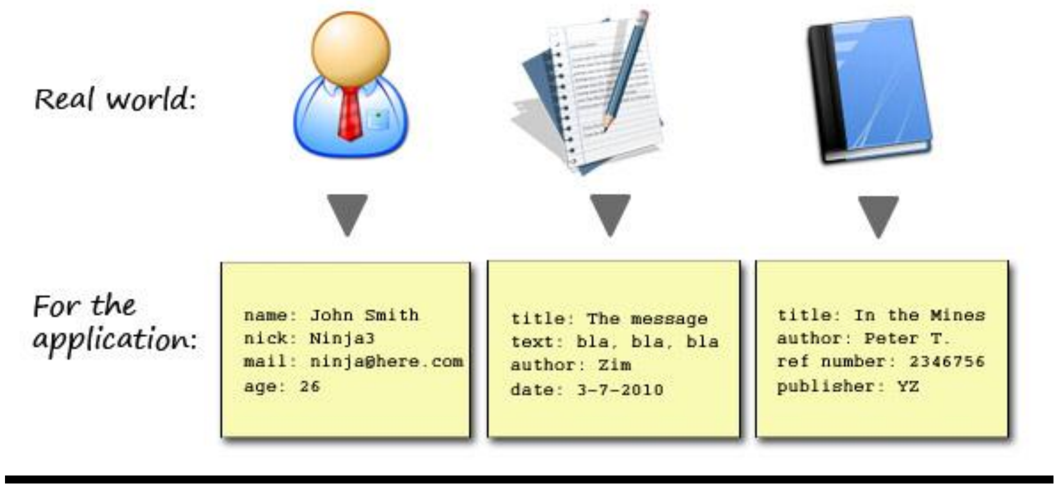
The model gives the controller a data representation of whatever the user requested (a message, a list of books, a photo album, etc.). This data model will be the same no matter how we may want to present it to the user, that's why we can choose any available view to render it.

The model contains the most important part of our application logic, the logic that applies to the problem we are dealing with (a forum, a shop, a bank, etc.). The controller contains a more internal-organizational logic for the application itself (more like housekeeping).

- A model is an object representing data or even activity, e.g. a database table or even some plant-floor production-machine process.
- The model manages the behavior and data of the application domain, responds to requests for information about its state and responds to instructions to change state.

- The model represents enterprise data and the business rules that govern access to and updates of this data. Often the model serves as a software approximation to a real-world process, so simple real-world modeling techniques apply when defining the model.

- The model is the piece that represents the state and low-level behavior of the component. It manages the state and conducts all transformations on that state. The model has no specific knowledge of either its controllers or its views. The view is the piece that manages the visual display of the state represented by the model. A model can have more than one view.



View

The **View** provides different ways to present the data received from the model. They may be templates where that data is filled. There may be several different views and the controller has to decide which one to use.

A web application is usually composed of a set of controllers, models and views. The controller may be structured as a main controller that receives all requests and calls specific controllers that handles actions for each case.

- A view is some form of visualization of the state of the model.
- The view manages the graphical and/or textual output to the portion of the bitmapped display that is allocated to its application. Instead of a bitmapped display the view may generate HTML or PDF output.
- The view renders the contents of a model. It accesses enterprise data through the model and specifies how that data should be presented.
- The view is responsible for mapping graphics onto a device. A view typically has a one to one correspondence with a display surface and knows how to render to it. A view attaches to a model and renders its contents to the display surface.

Chapter 2. Feasibility

In this chapter you will learn:

1. Technical Feasibility

Technical Feasibility

At the time of developing the project LaraComm, I have faced many hurdles. Hurdles like, UI (User Interface), back end problem, server error, browser problem, etc.

So, it was decided to opt for the already built 3rd party plugins and/or extensions. The plugins and extensions that are used in this project are as follows:

1. Toastr – Notification plugin to the user.
2. Cart – Cart plugin to add / update / remove products from the cart.

Chapter 3. Technological Aspect

In this chapter, you will learn about:

1. Technologies required on the client side.

2. Technologies that were required to build the project.

Technologies required

Following are the requirements in order to access the web application:

Operating System	Windows, Linux, Mac, Android
RAM	1 GB, 2GB Recommended
Browser	Chrome 4.0+, Mozilla Firefox 30+
Internet	YES

Technologies Used

Following are the technologies that were used to build the application:

Language	PHP – Laravel 5.0.32 Framework
Database	MySQL
IDE	PHPStorm 8.0.2
Internet	YES

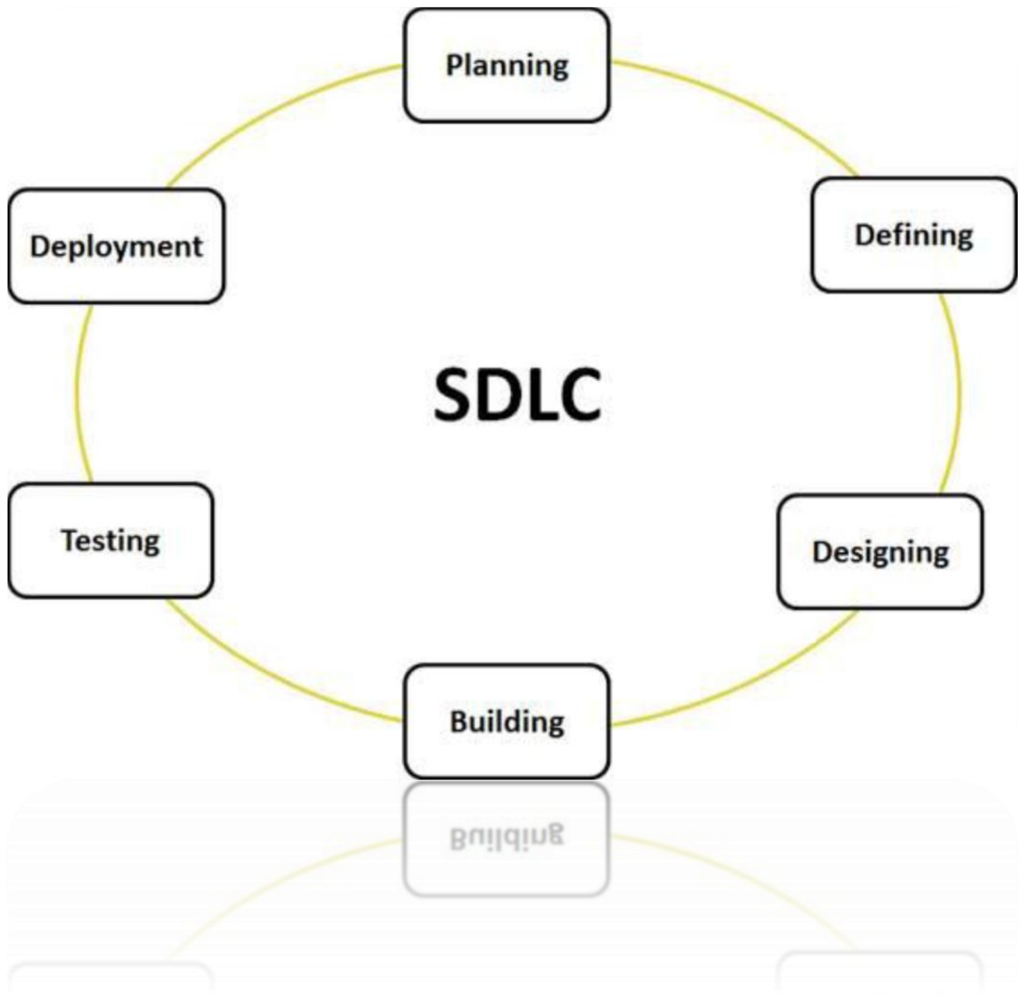
Software Development Life Cycle (S. D. L. C)

In this chapter, you will learn about:

1. What is S. D. L. C?
2. Various Models in S. D. L. C.

What is S. D. L. C.?

SDLC is the acronym of Software Development Life Cycle. It is also called as Software development process. The software development life cycle (SDLC) is a Framework defining tasks performed at each step in the software development process. ISO/IEC 12207 is an international standard for software life cycle processes. It aims to be the standard that defines all the tasks required for developing and maintaining. SDLC is a process followed for a software project, within a software organization. It consists of a detailed plan describing how to develop, maintain, replace and alter or enhance specific software. The life cycle defines a methodology for improving the quality of software and the overall development process.



<u>A typical Software Development life cycle consists of the following stages:</u>

Stage 1: Planning and Requirement Analysis: Requirement analysis is the most important and fundamental stage in SDLC. It is performed by the senior members of the team with inputs from the customer, the sales department, market surveys and domain experts in the industry. This information is then used to plan the basic project approach and to conduct product feasibility study in the economical, operational, and technical areas.

Stage 2: Defining Requirements: Once the requirement analysis is done the next step is to clearly define and document the product requirements and get them approved from the customer or the market analysts. This is done through 'SRS' – Software Requirement Specification document which consists of all the product requirements to be designed and developed during the project life cycle.

Stage 3: Designing the product architecture: SRS is the reference for product architects to come out with the best architecture for the product to be developed. Based on the requirements specified in SRS, usually more than one design approach for the product architecture is proposed and documented in a DDS - Design Document Specification. This DDS is reviewed by all the important stakeholders and based on various parameters as risk assessment, product robustness, design modularity , budget and time constraints , the best design approach is selected for the product.

Stage 4: Building or Developing the Product: In this stage of SDLC the actual development starts and the product is built. The programming code is generated as per DDS during this stage. If the design is performed in a detailed and organized manner, code generation can be accomplished without much hassle.

Stage 5: Testing the Product: This stage is usually a subset of all the stages as in the modern SDLC models, the testing activities are mostly involved in all the stages of SDLC. However this stage refers to the testing only stage of the product where products defects are reported, tracked, fixed and retested, until the product reaches the quality standards defined in the SRS.

Stage 6: Deployment in the Market and Maintenance: Once the product is tested and ready to be deployed it is released formally in the appropriate market. Sometime product deployment happens in stages as per the organizations' business strategy. The product may first be released in a limited segment and tested in the real business environment (UAT- User acceptance testing).

SDLC Models

There are various software development life cycle models defined and designed which are followed during software development process.

- Waterfall Model
- Iterative Model
- Spiral Model
- V-Model
- Big Bang Model

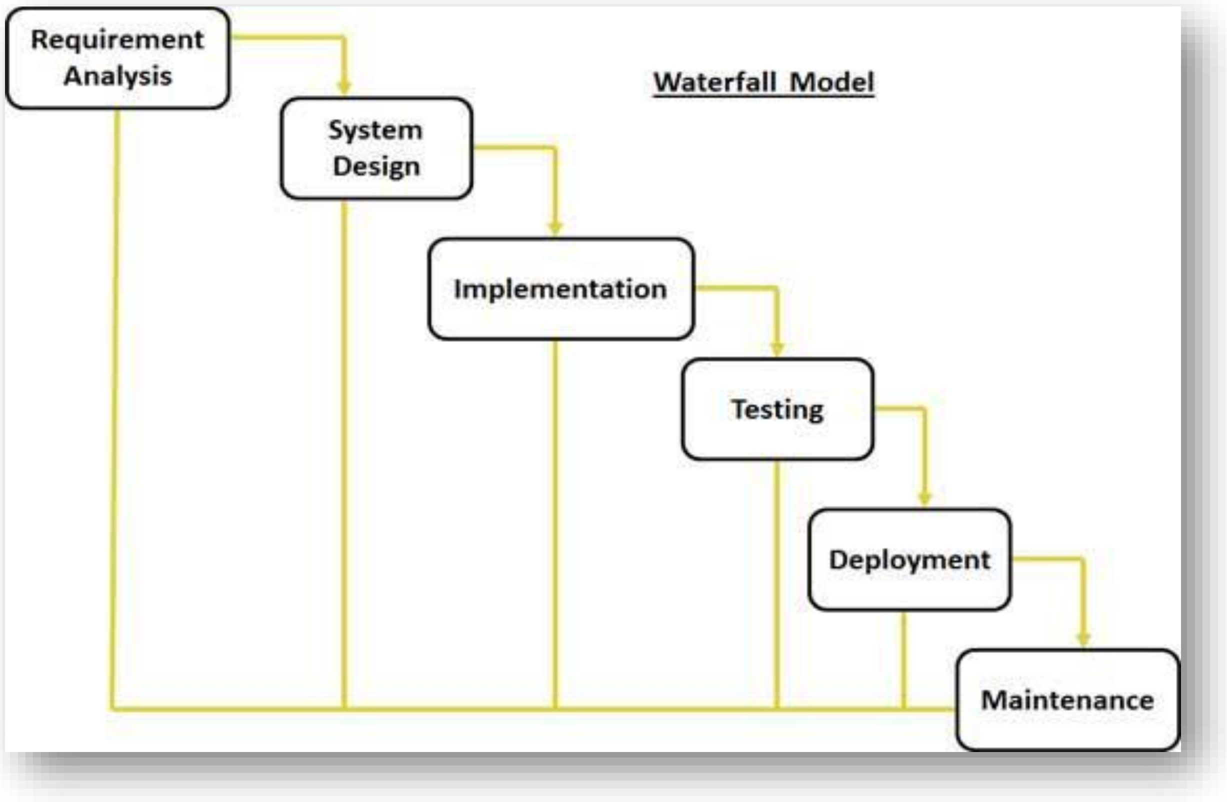
The other related methodologies are Agile Model, RAD Model – Rapid Application Development and Prototyping Models.

Waterfall Model

Waterfall model is the earliest SDLC approach that was used for software development .The waterfall Model illustrates the software development process in a linear sequential flow; hence it is also referred to as a linear-sequential life cycle model. This means that any phase in the development process begins only if the previous phase is complete. In waterfall model phases do not overlap.

Waterfall Model design

Waterfall approach was first SDLC Model to be used widely in Software Engineering to ensure success of the project. In "The Waterfall" approach, the whole process of software development is divided into separate phases. In Waterfall model, typically, the outcome of one phase acts as the input for the next phase sequentially.
Following is a diagrammatic representation of different phases of waterfall model.



The sequential phases in Waterfall model are:

- **Requirement Gathering and analysis** All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification doc.

- **System Design:** The requirement specifications from first phase are studied in this phase and system design is prepared. System Design helps in specifying hardware and system requirements and also helps in defining overall system architecture.

- **Implementation:** With inputs from system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality which is referred to as Unit Testing.

- **Integration and Testing:** All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.

- **Deployment of system:** Once the functional and non-functional testing is done, the product is deployed in the customer environment or released into the market.

- **Maintenance:** There are some issues which come up in the client environment. To fix those issues patches are released. Also to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

All these phases are cascaded to each other in which progress is seen as flowing steadily downwards (like a waterfall) through the phases. The next phase is started only after the defined set of goals are achieved for previous phase and it is signed off, so the name "Waterfall Model". In this model phases do not overlap. .

Waterfall Model Application

Every software developed is different and requires a suitable SDLC approach to be followed based on the internal and external factors. Some situations where the use of Waterfall model is most appropriate are:

- Requirements are very well documented, clear and fixed
- Product definition is stable
- Technology is understood and is not dynamic
- There are no ambiguous requirements
- Ample resources with required expertise are available to support the product
- The project is short

Waterfall Model Pros & Cons

The advantage of waterfall development is that it allows for departmentalization and control. A schedule can be set with deadlines for each stage of development and a product can proceed through the development process model phases one by one. Development moves from concept, through design, implementation, testing, installation, troubleshooting, and ends up at operation and maintenance. Each phase of development proceeds in strict order.

The disadvantage of waterfall development is that it does not allow for much reflection or revision. Once an application is in the testing stage, it is very difficult to go back and change something that was not well-documented or thought upon in the concept stage.

Pros	Cons
• Simple and easy to understand and use. • Easy to manage due to the rigidity of the model – each phase has specific deliverables and a review process. • Phases are processed and completed one at a time. • Works well for smaller projects where requirements are very well understood. • Process and results are well documented. • Clearly defined stages • Easy to arrange tasks. • Well understood milestones.	• No working software is produced until late during the life cycle. • High amounts of risk and uncertainty. • Not a good model for complex and object-oriented projects. • Poor model for long and ongoing projects. • Not suitable for the projects where requirements are at a moderate to high risk of changing. So risk and uncertainty is high with this process model. • It is difficult to measure progress within stages. • Cannot accommodate changing requirements. • No working software is produced until late in the life cycle. • Adjusting scope during the life cycle can end a project • Integration is done as a "big-bang" at the very end, which doesn't allow identifying any technological or business bottleneck or challenges early.

Chapter 5: Diagrams

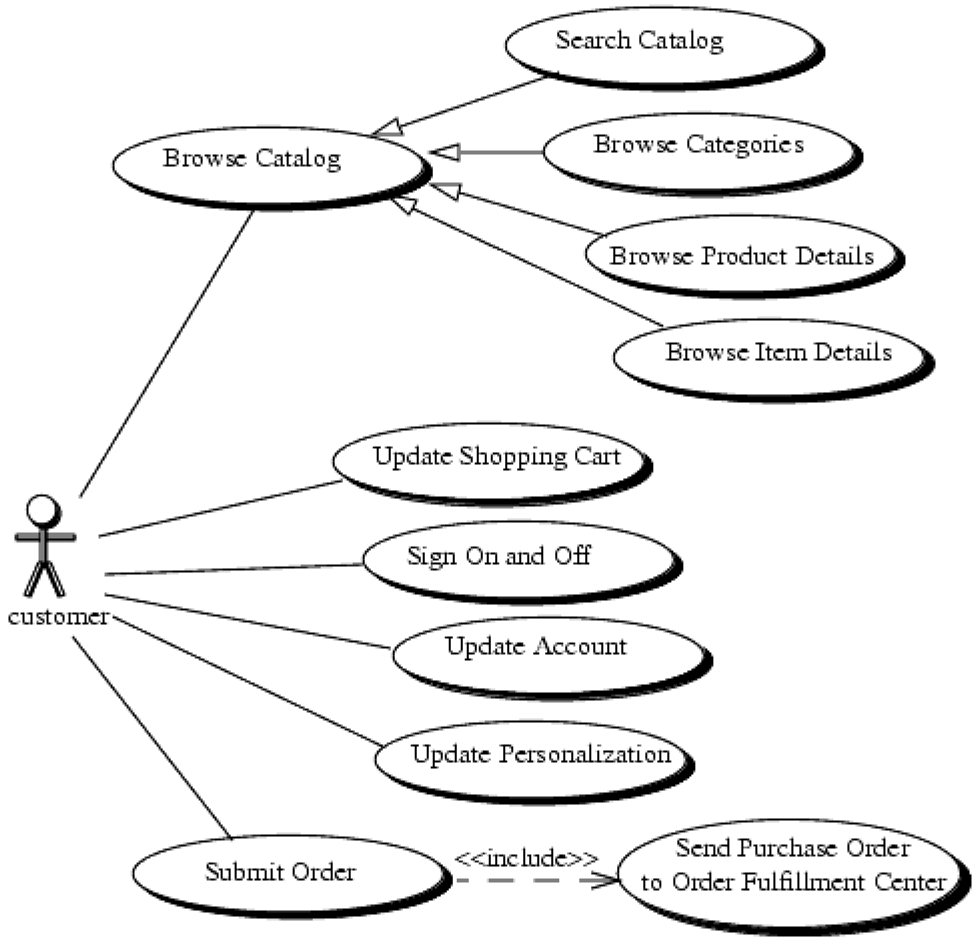
In this chapter you will see the diagrams:

1. Customer Use Case Diagram

2. Data Flow Diagram

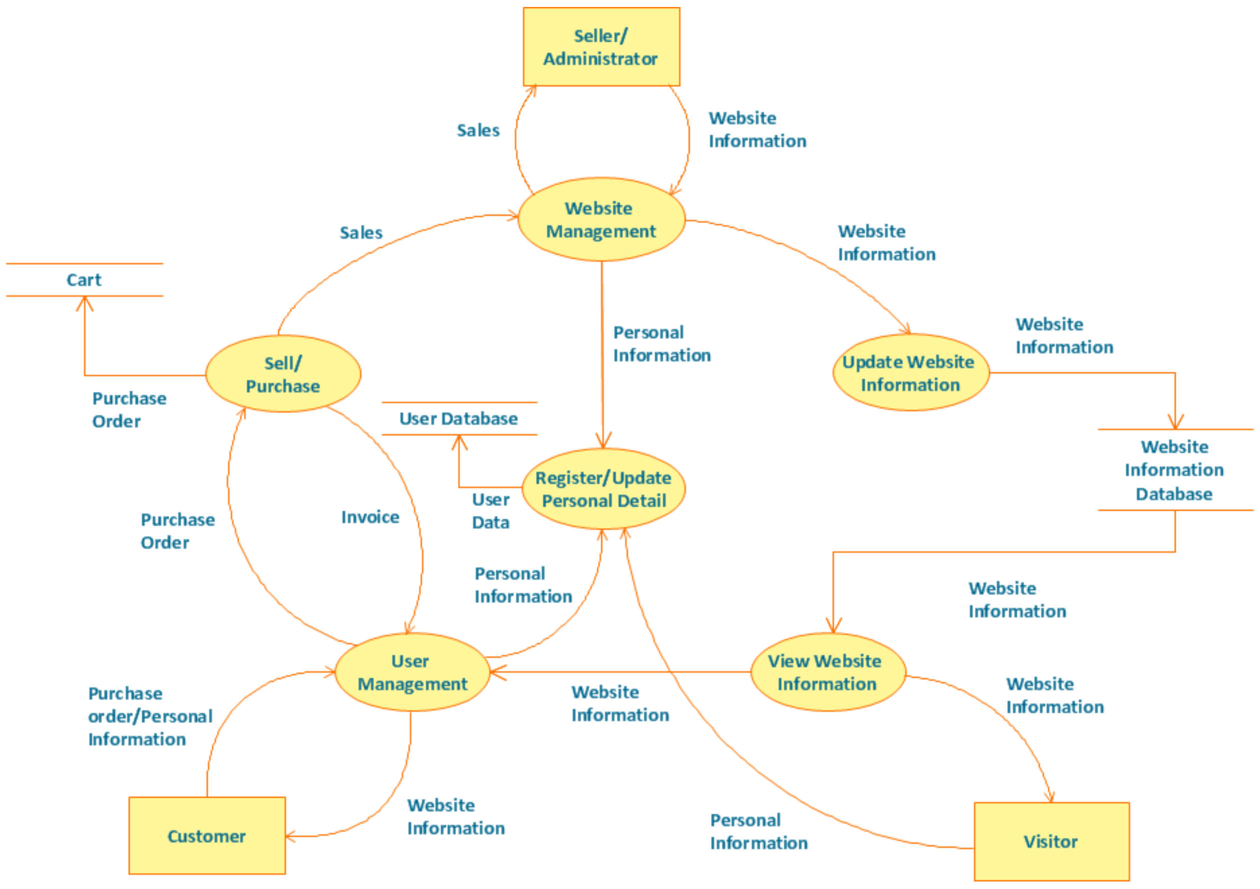
Customer Use Case Diagram

The use case diagram of the customer in the application



Application Data Flow Diagram

This is how the data will flow in the application.



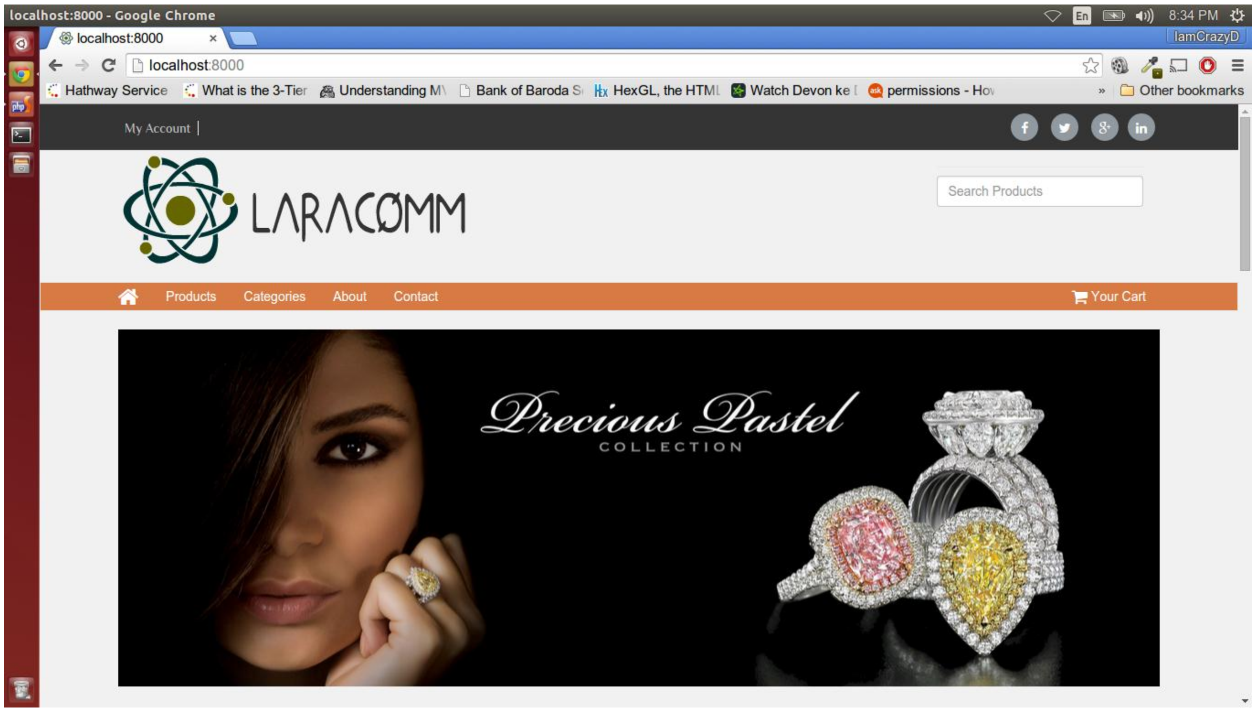
Chapter 6. Screen Shots

In this chapter you will see:

1. Application's Screen Shots

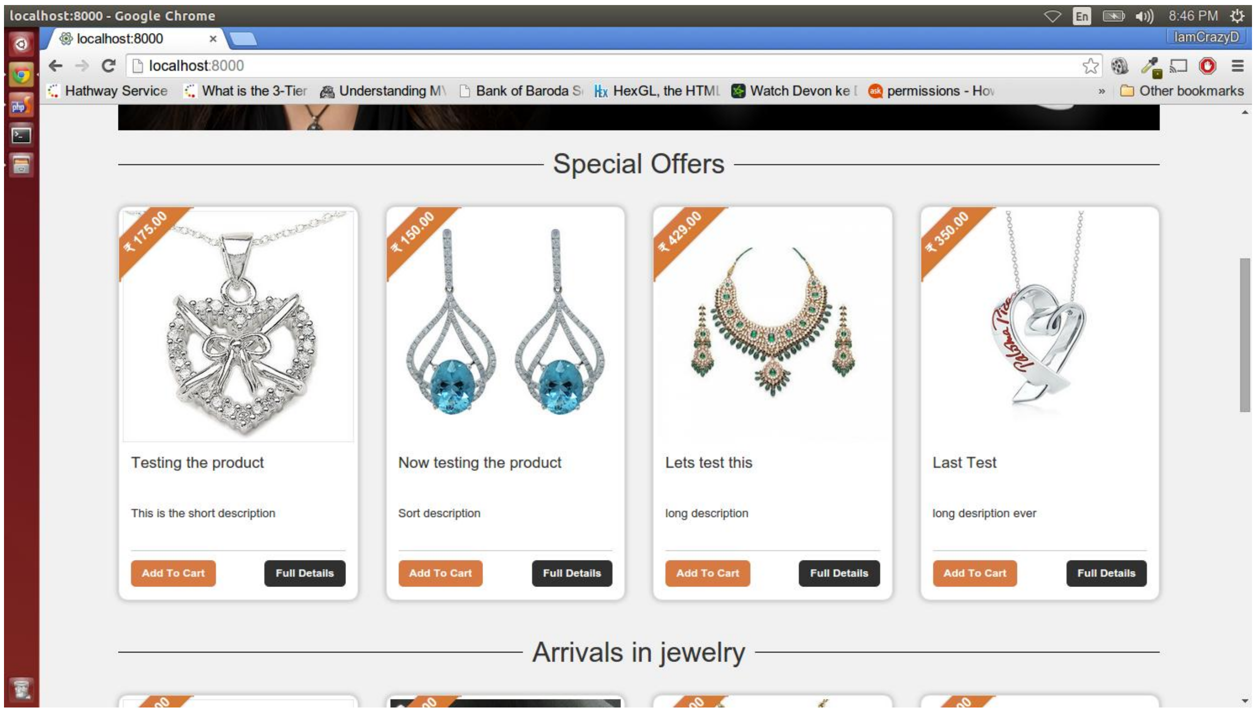
Screen Shots

The Header section on the home page



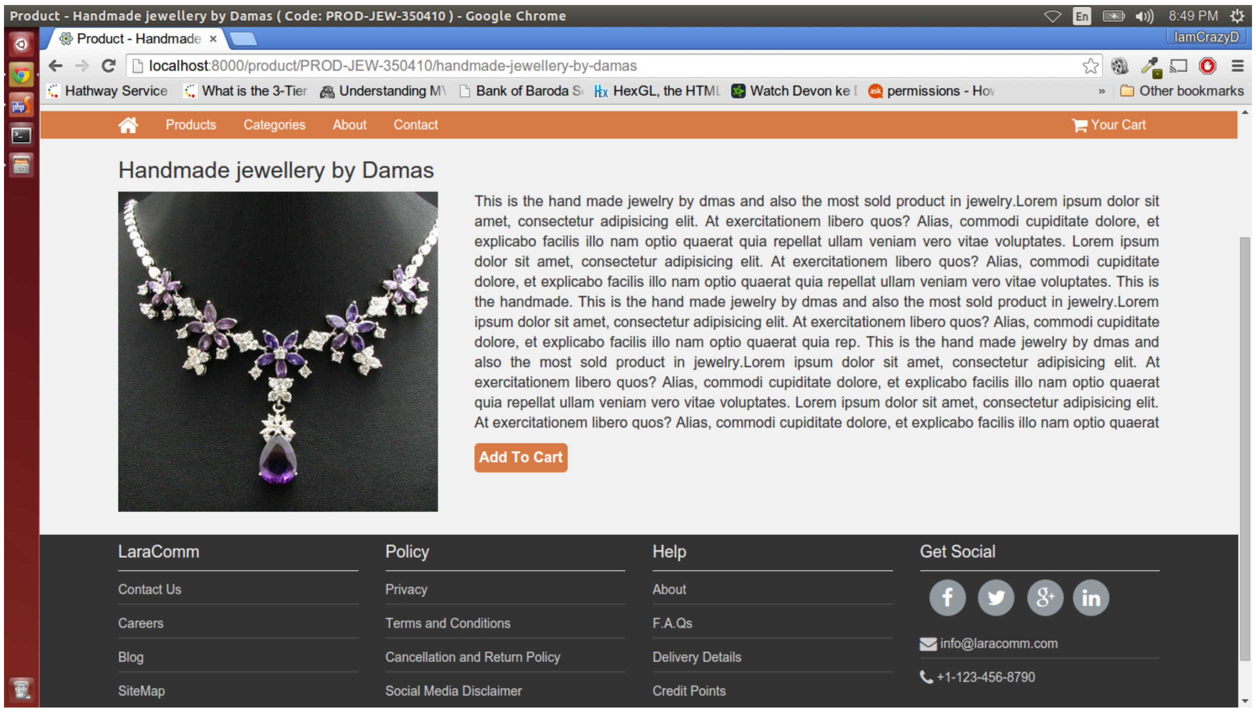
This is the header part of the application when the user lands on the home page.

The products Section on the Home Page



These are the products that are shown on the home page.

The Product Page



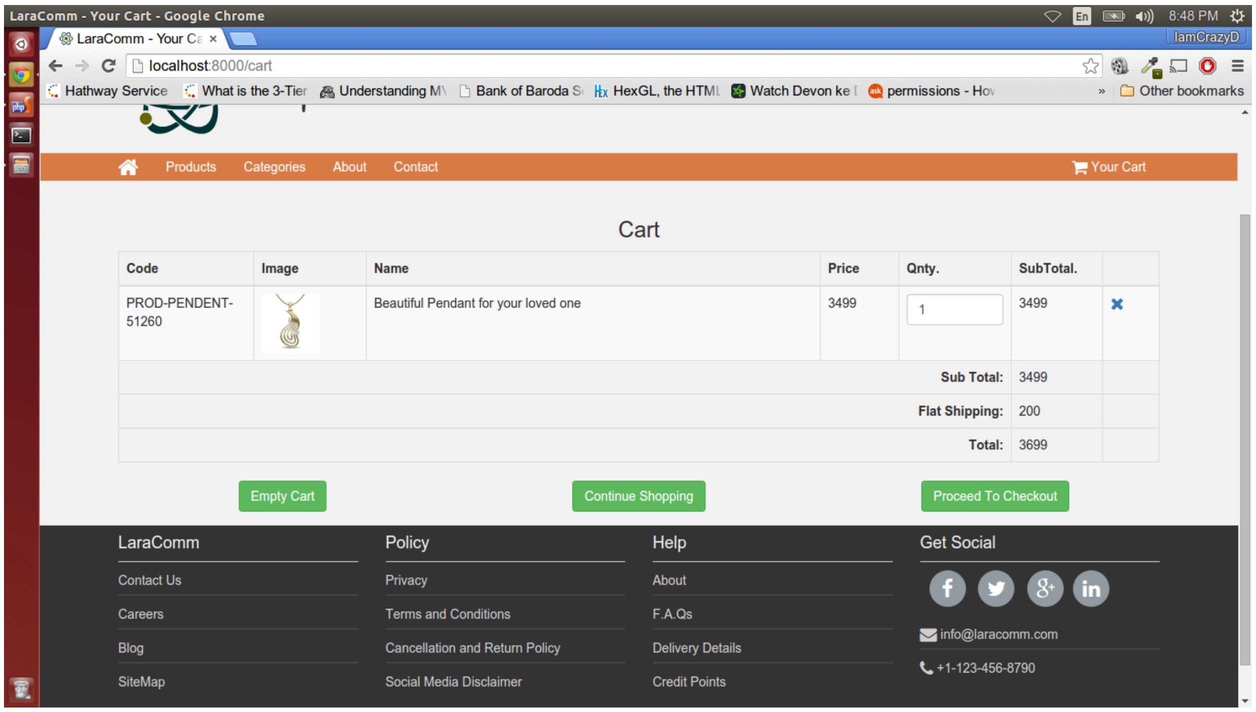
Handmade jewellery by Damas

This is the hand made jewelry by dmas and also the most sold product in jewelry.Lorem ipsum dolor sit amet, consectetur adipisicing elit. At exercitationem libero quos? Alias, commodi cupiditate dolore, et explicabo facilis illo nam optio quaerat quia repellat ullam veniam vero vitae voluptates. Lorem ipsum dolor sit amet, consectetur adipisicing elit. At exercitationem libero quos? Alias, commodi cupiditate dolore, et explicabo facilis illo nam optio quaerat quia repellat ullam veniam vero vitae voluptates. This is the handmade. This is the hand made jewelry by dmas and also the most sold product in jewelry.Lorem ipsum dolor sit amet, consectetur adipisicing elit. At exercitationem libero quos? Alias, commodi cupiditate dolore, et explicabo facilis illo nam optio quaerat quia rep. This is the hand made jewelry by dmas and also the most sold product in jewelry.Lorem ipsum dolor sit amet, consectetur adipisicing elit. At exercitationem libero quos? Alias, commodi cupiditate dolore, et explicabo facilis illo nam optio quaerat quia repellat ullam veniam vero vitae voluptates. Lorem ipsum dolor sit amet, consectetur adipisicing elit. At exercitationem libero quos? Alias, commodi cupiditate dolore, et explicabo facilis illo nam optio quaerat

LaraComm	Policy	Help	Get Social
Contact Us	Privacy	About	
Careers	Terms and Conditions	F.A.Qs	info@laracomm.com
Blog	Cancellation and Return Policy	Delivery Details	+1-123-456-8790
SiteMap	Social Media Disclaimer	Credit Points	

This is the product page when the user clicks on the “***<u>Full Details</u>***” button on the home page.

The Cart Page



Cart

Code	Image	Name	Price	Qnty.	SubTotal.	
PROD-PENDENT-51260		Beautiful Pendant for your loved one	3499	1	3499	✖
				Sub Total:	3499	
				Flat Shipping:	200	
				Total:	3699	

LaraComm: Contact Us, Careers, Blog, SiteMap

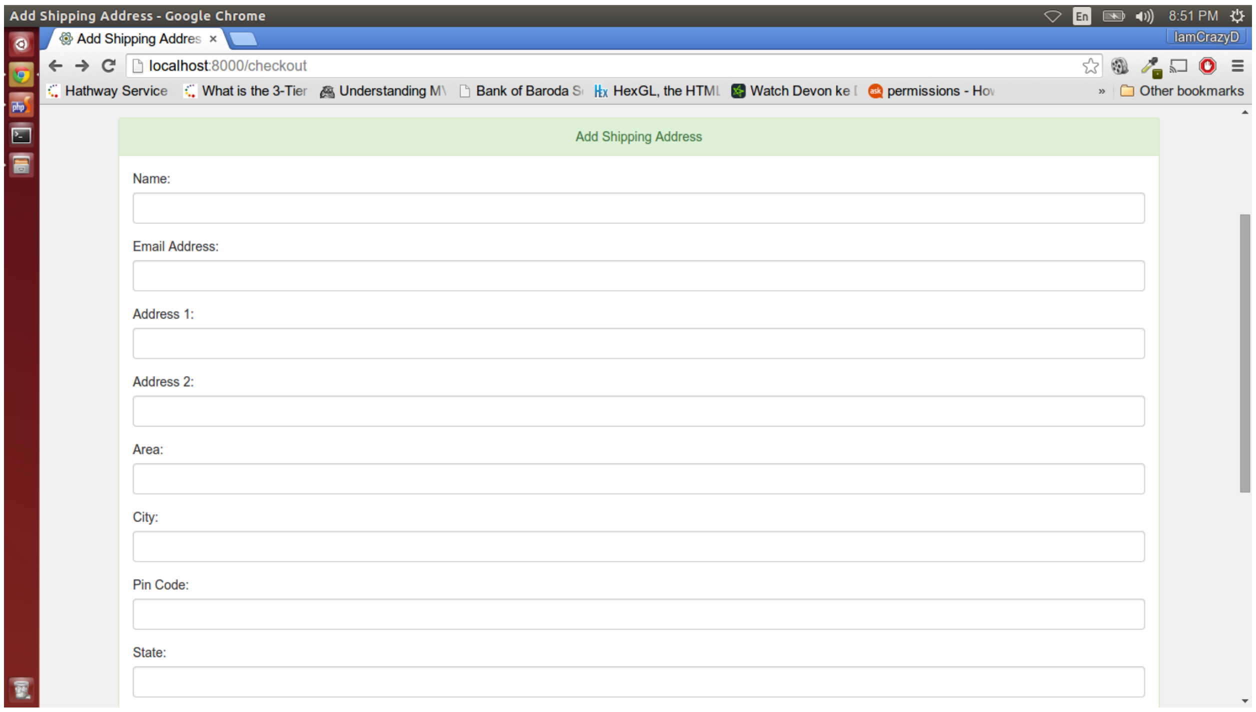
Policy: Privacy, Terms and Conditions, Cancellation and Return Policy, Social Media Disclaimer

Help: About, F.A.Qs, Delivery Details, Credit Points

Get Social: info@laracomm.com, +1-123-456-8790

This is the cart page. When the user clicks on the "***<u>Add to Cart</u>***" button on the home page or on the product page, he will see the cart here.

The Shipping Address Information Page



Add Shipping Address

Name:

Email Address:

Address 1:

Address 2:

Area:

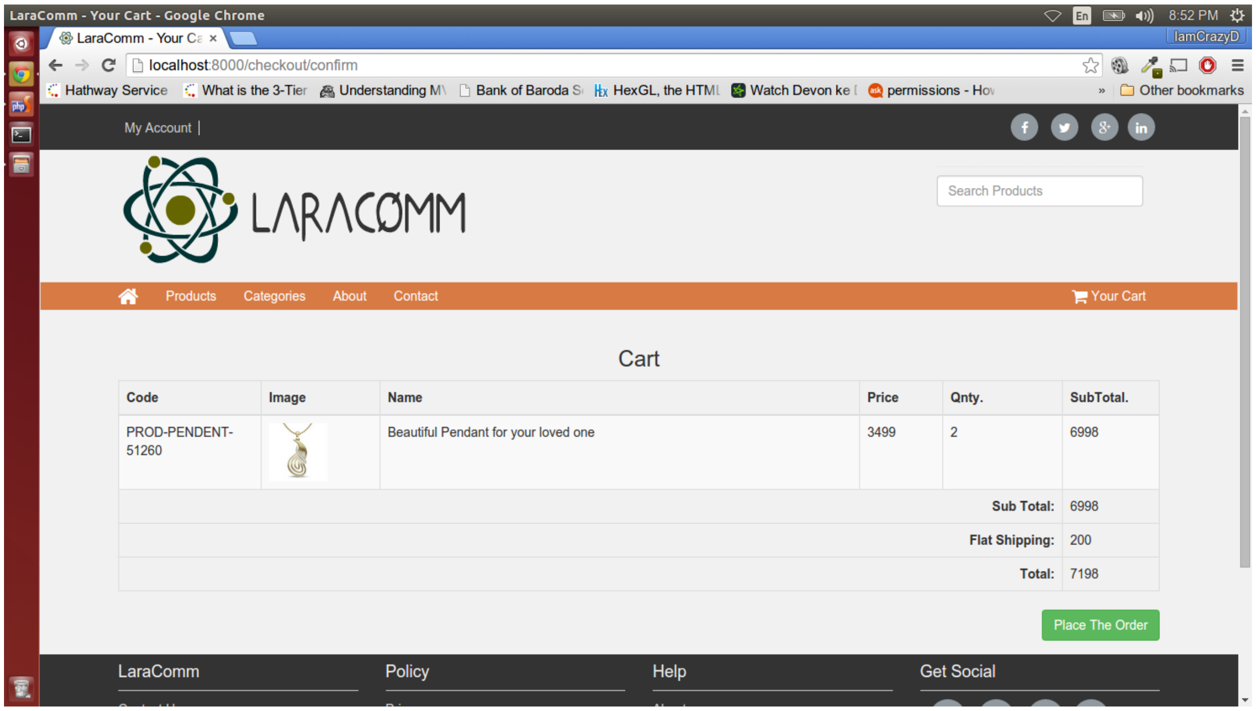
City:

Pin Code:

State:

The user is will see this page where he will have to add the shipping delivery address.

The Confirm Cart Page



Cart

Code	Image	Name	Price	Qnty.	SubTotal.
PROD-PENDENT-51260		Beautiful Pendant for your loved one	3499	2	6998
				Sub Total:	6998
				Flat Shipping:	200
				Total:	7198

Here he just needs to confirm the cart and click on the “***<u>Place the Order</u>***” button.

Chapter 7. Conclusion

In this chapter, you will learn:

1. Revise whatever we have gone through.

Revision of the application

Things you have studied so far:

1. What is E-Commerce?
2. Brief history of E-Commerce.
3. What Is MVC?
4. Study of Model in depth
5. Study of View in depth
6. Study of Controller in depth
7. What is LaraComm?
8. Waterfall model in detail.

Chapter 8: Future Enhancements

In this chapter, you will learn:

1. The future upcoming releases and its features.

Future Releases and Enhancements

Following are the points that are considered for Version 0.2 which will be released by the end of the year, i.e. by 31st December, 2015.

1. Enhance the UI for the customer
2. Enhancing the UI for the administrator
3. Addition of more products in multiple categories
4. Support for I.E. <= 8
5. Carousels with more than 4 products
6. Making it live on the internet
7. Making the source code live on GitHub
8. Making the mobile application for Android and iOS

Chapter 9: Bibliography

In this, you will:

1. See the references

Bibliography

Following are the sources that helped me to build this whole application:

1. http://www.laravel.com/docs/5.0
2. http://laracasts.com/series
3. http://stackoverflow.com
4. http://programmers.stackexchange.com
5. http://codereview.stackexchange.com