Final Year Thesis

Convenient, expendable and secure Social-Commerce web platform for small businesses and consumers

Stefan Grulović

Supervisor: Mr. Hristos Hristodoulou
Division of Computer Science
American College of Thessaloniki Pilea, Greece

Contact: stefan.grulovic@gmail.com

Abstract

The project involved developing a social-commerce web application which intends to be convenient, scalable and secure providing businesses and consumers with a platform through which they could communicate and trade.

The main goal of the application is to provide a convenient platform where businesses and consumers can create a profile to simply and directly establish a connection between them. Supplying a profile through which businesses can promote their services and sell their products while consumers have the ability to simply search and compare those findings to a more suitable offer.

The second goal is to create an expandable platform which allows further growth of the application. This is achieved by using Laravel framework, MVC design pattern and object orientation principles which provide the necessary organization.

Lastly, the project aims to create a secure platform in order to survive remotely in real word scenario. This is vital due to security concerns of money being involved.

Validation conducted on the application tested the user experience and actual performance of these three goals. User experience was validated using surveys given to businesses, consumers and programmers for checking if it is convenient, easy to use, expandable and secure. Additionally, the performance of the application was put to test using database simulations, SQL Stress and Load Tests and RAIL Model.

Acknowledgements

I am grateful to the instructors of the Computer Science Department for their generous training, assistance, and guidance. I thank my classmates and roommates for all of their assistance and support. I also would like to thank the computer science faculty for their constant provision and encouragement in efforts to enhance computer science education at American College of Thessaloniki.

.

Table of Contents

Abstract i

Acknowledgements ii

Acronyms v

1 Introduction 6

1.1 Motivation for the project 6

1.1.1 Stating the problem 7

1.2 Project objectives 7

1.2.1 Business specific objectives 8

1.2.2 Consumer specific objectives 8

1.3 Literature review 9

1.4 Existing state of the art solutions 10

2 Proposed design Requirements and Timeline 12

2.1 Extracting the design requirements 12

2.1.1 Example scenarios 12

2.1.2 Use cases 14

2.1.3 Functional requirements 20

2.1.4 Non-functional requirements 27

2.1.5 Software Design Specifications 28

2.2 List of design requirements 34

2.3 Development milestone 35

2.4 Proposed timeline 36

2.5 Contingency planning 37

3 Description of hardware 39

4 Description of software 40

4.1 Laravel 40

4.1.1 Artisan 41

4.1.2 Schema, Migrations 41

4.1.3 Eloquent ORM 41

4.1.4 Blade 41

4.1.5 Routes and Controllers 42

4.1.6 Security 42

4.2 Bootstrap 43

4.3 jQuery 44

5 Planned validation experiments 45

5.1 Load Testing 45
5.2 Stress Testing 45
5.3 RAIL Model 46
5.4 Results 47
5.4.1 Survey Results 47
5.4.2 Stress and Load Tests Results 50
5.4.3 RAIL Model / Lighthouse (Google Chrome Audits) Results 51
5.4.4 Software Metrics (PHPMetrics) 52
6 A look into the future 54
6.1 Advances in the technologies relevant to the project 54
6.2 Future improvements (on this project) 54
6.3 Insights into future applications 55
Appendices 56
Glossary 62
Table of figures 62
Table of tables 62
Table of diagrams 62
Table of picture 62
Bibliography 64

Acronyms

MIT	Michigan Institute of Technology
MVC	Model-View-Controller
PC	Personal Computer
HTML	Hypertext Markup Language
CSS	Cascading Style Sheet
SQL	Standardized Query Language
PL/SQL	Procedural Language/Structured Query Language
PHP	Hypertext Preprocessor
RAIL	Response, Animation, Idle and Load
AJAX	Asynchronous JavaScript and XML
PDO	PHP Data Objects
JS	JavaScript
GUI	Graphical User Interface

1 Introduction

It is possible to create social-commerce web platform, which is convenient and simple to use, expandable and easy to maintain and secure providing businesses and consumers with a platform for communication and trade.

Hermes is a god from Ancient Greek religion and mythology who was in charge of trade, commerce, and sports. A messenger of the gods with many other duties. Being the god of trade provides the commerce aspect, but also a messenger and in charge of social activities such as sports, which provides the social aspects, making it a suitable choice for the name.

Picture 1. *The logo of the Social-Commerce platform*

1.1 Motivation for the project

With the growth of social media, the use and integration of social media became vital for today's businesses. This project combines the e-commerce and social media aspects into one, called social-commerce, in order to create a convenient platform for business and consumer to connect. [1] It is an idea that goes beyond social media and commerce platforms but one with the functionality of both. This project is also capable of displaying skills learned in the bachelor's degree, mainly focusing on programming skills, software and web development. [2]

1.1.1 Stating the problem

Nowadays we see more examples of social media trying to adapt elements of commerce rather than other way around. [3] The project provides a social-commerce platform where the commerce gets its own version of a social platform, where the primary focus is the commerce and social aspect of the exchange.

1.2 Project objectives

- **Convenient, simple and easy to use**

  The platform is tailored in such a way that an individual with no background in computers should be able to navigate with ease which is needed to ensure user satisfaction. [4] [5]

- **Easily expandable**

  By using object orientation programming, design patterns and good coding practices, a simple and clear code and documentation is created. Providing the organization and information needed in order to easily expand and further improve the platform.

- **Secure**

  The security of the platform is a primary concern to survive in a real-world scenario. The application keeps the private information protected by encrypting and/or hiding the information from other uses from being able to add, see, modify and or delete.

- **Connecting the businesses and or consumers with social media aspects**

  Social features that the platform provides are the ability to easily search and add friends, store favorite users/pages, directly message and engage in open forum discussions.

- **Molding of the platform experience to user's preferences and usage**

  The information from user's searches, comparisons, ratings and transactions could be used to improve platform usage towards the user preferences.

- Statistical data for businesses

The data could be used to create statistical data on businesses profiles, products and services. Providing information on how they are doing, the traffic which they are getting and so on.

- Recommendation for consumers

On the other hand, the information which the consumers produce with their usage of the application could be used to create better recommendations for consumers.

1.2.1 Business specific objectives

- **Appealing profile for businesses**

Profile gives the businesses the ability to create a page where they can add, promote and sell their offers. The page displays everything they have to offer which consumers can access and make an order.

- **Ability to track and respond the consumer requests**

Keeps track of all the requests from the consumers and also allows searching for specific request by the name of offers or users, status of the requests and payment type. The businesses are also able to display each request and see in convenient table all the information about the specific request as well as the ability to directly respond to the user the status of request, shipping and delivery dates.

1.2.2 Consumer specific objectives

- **Search for products and services**

The consumer is able to search through users, offers and forum threads in order to quickly find what it is that they are looking for. When searching for users, one can search them either by name, email, profile type and category. When searching the offers, one can search again by categories, maximum price and to whom the offer belongs to. For forum threads, one can search by the name of the thread or the user, by whom the thread belongs to.

- **Cart feature**

A cart feature gives consumers the ability to quickly store offers which they would like to order as well as send the order to the corresponding business.

- **Compare feature**

A compare feature gives consumers the ability to immediately compare products by adding them to the stack in which they can also be sorted based on all of their information. Quickly comparing and contrasting products and/or services one can find faster what suits them best.

- **Comments and Ratings**

The consumer can add a comment and rating to any offer the businesses provide. This way the consumer can inform both the business and consumers with the feedback of how they have experienced their product or services.

1.3 Literature review

Since the invention of the internet, the ability to buy and sell product online was made possible. Nowadays there are numerous attempts and platforms which try to accomplish that. E-commerce are online services which allow the buying and selling of products and or services. [7] For e-commerce there are a lot of examples of well-developed platforms such as Amazon, eBay, Ali-baba etc. However, most of the e-commerce platforms do not include the social media aspects into their application; this is where social-commerce comes into place. Since 2005 the term social-commerce is introduced by Yahoo! as a sub section of e-commerce. [8] Social-commerce is where e-commerce and social media aspects get merged into one platform in order to further support the exchange. However, for social-commerce there are fewer known examples compared to e-commerce and some of them are Betabrand, Etsy, Cafepress etc. [9] [10] [10] [2] [11]

Making the platform simple in use for both businesses and consumers would be vital in order to positively influence the use of the application. [4] [12] [13]

The platform is written using a framework called Laravel which allows development to be more efficient, and also incorporates an MVC design pattern to allow the necessary

organization to be ensured. [14] [15] The framework made the development even feasible for the given time and circumstances for one person to develop it. It also ensures and is the indicator of up to date industry standards as it is licensed by MIT, making it suitable for commercial use.

Making effective use of customer knowledge can give the application a competitive advantage. [5] This includes both the recommending system which would offer products based on their previous actions and purchases as well as the statistics about businesses profiles and their products or services. [16] [17] [18] [19]

With security being one of the most important factors in electronic commerce, it is an important element to properly take care off. [20] If rapid development is in use, the chances of vulnerability increases. [21] If the users do not feel the platform is trustworthy, they will be less willing to conduct business in the first place. [22] [23] [24] [25] [26]

The benefits of this project are to further define social-commerce as well as providing an actual example of such a platform. The project is also a benefit to businesses and consumers by connecting them through the platform, allowing businesses to sell more and consumers to make better purchases while communicating as one community. [10]

1.4 Existing state of the art solutions

Most state-of-the-art solutions focus primarily on the e-commerce aspect and less on social media integration. [3] This is because implementing one to the other is a costly action which could have a negative impact on the business and is highly undesirable by company owners. Because this project was created from scratch, it has the ability to easily introduce different aspects and combine them into one. The social-commerce is now mainly conducted indirectly in major social media platforms where they are linked to the businesses in some shape or form. [27] [28] Social-commerce should be the platform where a direct link is created among businesses and consumers to create a

single and concise experience. Most state-of-the-art solutions regarding social-commerce do not have much of the social media functionality and or is done in such a way that does not indicate that such option is available due to the complexity of how the platform navigation is designed. Previously mentioned examples of e-commerce and social-commerce in literature review are the current state of the art solutions.

2 Proposed design Requirements and Timeline

2.1 Extracting the design requirements

It should be pointed out that most of the social-commerce platforms do not have much of the social media aspects integrated or they are organized in complex manner, the main design requirement of the project was to change this. This is achieved by creating a platform with a solid balance of both e-commerce and social media functionalities and philosophies. [2]

Actors of the system are:

- **Stakeholder**: People who want to invest into development of the application
- **Primary**: Business and consumers which would use the application
- **Supporting**: Programmers which would further expand the application

2.1.1 Example scenarios

2.1.1.1 *Business example*

The application offers businesses a platform in which they can create a profile in order to further promote and make their product or service more accessible in regards of being discovered as well as having a direct communication with the consumer. After creating the profile, business would be able to add all the products and or services they could offer which would be publicly accessible by the registered consumers. The application also offers businesses a way to neatly keep, search and respond to requests made by the consumers. As well as social aspects where they can be contacted directly by the consumer and discuss whatever needs to be discussed as well as having forum threads on various topics on which they wish to gather information. The page also provides statistics where businesses can track progress over time. It is valuable information to know for the future of their business.

2.1.1.2 *Consumer example*

On the other side of the spectrum the application offers a platform to consumers who want to discover and find better offers for products or service as well as have the ability to directly communicate with the business. This is done by giving the consumers ability to search for offers on various different parameters, as well as the ability of comparing multiple offers at once. The application also gives the user ability of searching and keeping track for all the orders the users has made and also based on the status of order modify their purchase. The application also offers consumers social aspects such as adding friends, conversation/messaging system where they could contact businesses directly as well as other consumers with also ability to make group conversations and forum which they could engage in open discussions with or to the business or consumer.

2.1.1.3 *Interaction between Business and Consumer example*

The main example between the business and the consumer is the act of exchanging. The following list explains how a business and consumer user from the creation of their profile can connect with each other.

- The business creates a profile and adds an offer or offers which provide the necessary information and or pictures, becoming public to all the consumers.
- The consumer also creates its own profile searches for an offer.
    - The user can search the offers by name, businesses name, category, price, etc.
        - The user can open the offer and check more of the information about the offer.
            - User can also see reviews which have the comments and ratings.
        - User can put the offers being debated inside the compare option.
            - Offers are then put in a stack that can be sorted on all of their fields until the user finds what suits the best.
- Consumer can then place the offer or offers in the cart.
    - The consumer can choose the amounts and payment method and the order is then sent to the business.

- The consumer is redirected to the orders page where he can see the current order information and status.
    - Depending on the status the user may still modify the amounts and payment method.

- The business checks the requests from the consumer on the requests page.
    - The user may search the requests based on offers name, users name, status and payment.
    - Selecting the request, the business can see all the information about the order, the user which is requesting the offer or offers with amounts and payment type, status, shipping date and delivery date.
        - Depending on status, the business can respond to the consumer by changing the status of the request, shipping and delivery dates

.

2.1.2 Use cases

Due to the application being a web page, the use cases can be grouped based on the navigation options and organized in three groups general, social, commerce and consumer storage.

2.1.2.1 *General*

- **Welcome**
    - Shows options to Register and Login
        - To register or login, the user needs to provide all the information asked which would in return create or log into the profile.
- **Home**
    - For ***businesses*** display the statistic of how their page, product and or services are doing. Statistics such as number of offers per customer and offer, numbers of offers per month.
    - For ***consumers*** it would display the number of orders per category and recommendations based on their activity and purchases.

- **Documentation**
  - Shows the full documentation of the page.
- **About/Contact**
  - Shows the information about the page as well as the contact information.

2.1.2.2 *Social*

- **Friends (Consumers)**
  - Displays the list of friends.
    - Clicking from the list would display the user's profile and some of their activity such as their offers, forum, threads etc.
    - Hovering over one of the user's button to remove a friend is shown where the user may choose to remove a friendship.
  - Displays the friend requests.
    - By hovering over the request, the two buttons are shown where the user may either reject or approve a friend request.
- **Users**
  - Shows the full list of users both businesses and consumers.
    - The user may search other users based on their name, email, profile type etc.
    - Hovering over the one of the showed users' different options are shows such as:
      - Add friend.
        - This would send a request to the user.
      - Add to favorite.
        - This adds the user to the favorites.
      - Message user.
        - This creates a new direct conversation to the user and is redirected to the conversation page where the users may communicate.
      - Go to profile.
        - This would take the user to the businesses or consumers profile.

- Also clicking on the specific user shown the user is redirected to businesses page or consumers profile.

- **Conversations**
  - The user can search the conversation based on participant's names, emails or conversation name.
  - List of all conversations the user is participating are shown. Both direct and group conversations.
    - Clicking on one of the conversations a chat is shown with all its messages alongside the form for sending a new message.
      - Filling the message input form and clicking send a message is sent and posted.
      - Logged user messages are pushed on the right and other participants messages are pushed to the left alongside their avatar.
        - Clicking on message one can check the date and time the message was sent.
        - Clicking on own message alongside date and time the remove message option is shown.
          - Removing the message would replace the current message with a notification that the user has deleted its own message.
        - Clicking on avatar one is redirected to the user's page.
      - In case the user has created the conversation the options to edit and remove the conversation are shown.
        - In case the user clicks the edit the edit form is shown with the list of participants and conversation name.
          - Add participant by searching the users based on their name or email.
            - Adding a participant would send a notification to the conversation that a user has been added.
          - Remove participants from the conversation.

        - Removing a participant would send a notification to the conversation that a user has been removed.
    - In case the user clicks the remove conversation the conversation is removed.
- The user may also click Create a New Conversation which shows the form which has the conversation name and participants search bar.
    - The user can add all the users or user which he wants to create a new conversation with by searching users by name or email.
    - Optionally he can also add a name to the conversation where in that case the name would be shown in conversation list rather than all the users which are in the conversation.
- Hovering over the conversation list item the button to leave the conversation or delete conversation is shown.
    - If the user is creator of conversation delete conversation is shown otherwise leave conversation is shown.
        - In case the user leaves the conversation, a notification is sent to the conversation that the user has left the conversation.

- **Forum**
    - Shows the list of forum threads.
        - The user can search the threads based on threads name, creators name and or email, category.
        - Clicking on the thread from the list would show the thread and all of its replies alongside the form for adding a new reply and button for adding the thread into favorites.
            - In case the reply belongs to the current logged user a delete button is shown which gives the ability to delete a reply.
            - Filling the form message input and clicking send a reply is posted on the current thread.
        - Clicking Create New Thread a form is shown where the user needs to provide name, description and category for the thread.

- Filling the form and submitting it a new thread is created which is public to all the users.

2.1.2.3 *Commerce*

- **Offers**
  - Shows all the businesses offers
  - The user may search the offer based on their name, businesses name or email, category, price etc.
  - Hovering the offer the description is shown alongside buttons to:
    - Add offer to one of the user's lists.
    - Add or remove offer from favorites.
    - Add offer to cart list.
      - From where the user can chose the amount which it requires and payment method and clicking order would send the order to the business.
    - Add offer to compare stack.
      - From where the user can sort the offers based on all of their information's.
  - Clicking on the offer the user is taken to the offers page where all the information is shown alongside its reviews and ability to leave a review
    - In case the user wants to leave a review, the form needs to be filled with a comment and a rating from 1-5 stars and clicking on post review a review would be posted.
    - In case the user has posted a review, the review form is not shown, and a delete button is shown on the user's review.
    - In case the user is the creator of the page an Edit Offer button is shown.
      - Clicking on it takes the user to the edit form.
        - Where the user can edit the information of the offer such as name, description, price, available, unit type, images, category etc.
          - By filling the form user is shown a real time preview which is still local on the machine.
          - When finishing editing user should click the Update Offer which would update the information in the database.

        - Clicking on Remove Offer the offer is removed from the database.
- Clicking on Create New Offer takes the user to the create form page.
    - Filling the form by providing name, description, price, available, unit types, images, category and clicking submit a new offer is added to the database which is public to all users.

- **Requests (Businesses)**
    - Shows the list of requests the users have received which contain information from who is the request from, for what and when the request was made.
    - The business can search through the requests based on the name of the offers or the request user, request status and payment method.
    - Clicking on one of the orders the full order with all the information is shown which contains the date of order, status of order, for what offers and their amounts, payment method, shipping and delivery date.
        - The business can respond to the request by changing its status, adding or modifying the shipping and delivery date.
    - Based on different statuses the user can delete a request from the list.

- **Orders (Consumer)**
    - Shows the list of orders the users have made which contains information to whom the order is sent, for what and when.
    - The consumer can search through the orders based on the name of the offers name or its user, order status and payment method.
    - Clicking on one of the orders the full order with all the information is shown which contains the date of order, status of order, for what offers and their amounts, payment method, shipping and delivery date.
        - In case the order is in Ordered or Hold state the consumer can still modify the payment method and amounts of item.
    - Based on different statuses the user can delete an order from the list.

2.1.2.4 *Consumer Storage*

- **Lists (Consumers)**
  - Shows all the users lists and a button to show the form for creating a new list.
    - By hovering over a list item, a delete list button is shown.
    - Clicking on the list option shows all the lists offers.
      - With button on each list item, there is an option for removing the item from the list.
    - In case the user clicks create new list, a form is shown which requires name and description for creating a new list.
      - Clicking on create a new list is created with the inputted name and description.
- **Favorites (Consumers)**
  - Shows all the offers, users and threads that the user has favorited in a list form.
    - Clicking on one of the items shows the information of the given item.
      - Further clicking on the shown information, the user is redirected to the items page.
    - By hovering over one of the items, a delete button is shown which when clicked removes the item from the favorites.

2.1.3 Functional requirements

2.1.3.1 *General*

- Every user form submit should be validated whether it is in required form and format as well as if any malicious input has been added.
- Clicking the users, offers, threads name anywhere in the application where it was put by the application should redirect the corresponding item.

- **Welcome**
  - **Register**
    - After the user provides the needed information for registering, the system needs to create an entry for the user in the database.
      - Encrypt the user's password.
      - Then redirected to the Home page.
  - **Login**
    - After providing the needed information for logging in the system should store the current user and keep them logged in.
      - Redirect consumer to the Home page.
      - The user should be kept logged in while being active and logged out after certain amount of inactivity time.
- **Home**
  - **Business**
    - Calculate the necessary statistics from the database about the business and display to the user.
  - **Consumer**
    - Generate the recommended offers based on user searches and activity and also displays the chart of the activity.

2.1.3.2 *Social*

- **Friends**
  - Index page should retrieve friends from database and display all the user friends and friend requests in a list form.
    - Responding on the friend request with approve should update the status of requests statuses to approved for both entries in the database.
    - Responding with reject should delete the two request entries in friends table.
  - Clicking on a friend should retrieve the corresponding user public information alongside his offers and threads.

- **Users**
  - Index page should retrieve all the users and their public information from the database and display it to user.
  - If the users click the Search button the corresponding users with the similar name from the search input and selected profile type should be retrieved from the users table.
  - If the user clicks on the buttons:
    - Go to Profile
      - Should redirect user to the selected user's profile.
    - Add friend
      - Should create two entries in friends table with status "request" to the requested user and "rejected" to the
    - Add to favorites
      - Should create a new entry to the favorites table with the id and type "page".
    - Message user
      - Should create a new conversation with the user's id as a creator and two participants entries with the both of user's id with the newly create conversation id.
  - Every time the user updates the information about the profile on Users Edit it needs to be updated in the database.

- **Conversations**
  - Index page should display all the conversations that the user is participating from the database in a list from.
    - In case the conversation has a name, it would display the name otherwise display all the names of participants.
  - Using search to create a conversation form or edit conversation participants should display five similar users to the user's input.

- Clicking on a conversation should retrieve all the conversations participants and their messages.
    - Filling the message input and sending the information should create a new entry in messages table with the current conversation id and user's id.
    - Editing the conversation name should update the conversation name in the conversation table.
    - Editing the participants by either adding or removing a participant should add or remove a entry from the participants table.
        - And sending the corresponding alert to the conversation.
            - Either Participant Removed or Participant added.
    - The system should keep track of the latest message displayed and check every second if a new message was added and update the conversation with the new messages.

- **Forum**
- Index page should retrieve and paginate all the threads from the threads table and display it to user.
- Clicking on the thread from the list should retrieve the thread alongside its replies.
    - Clicking delete on its own thread reply the reply entry should be removed from the database.
    - Filling the send reply form and clicking the send button should respond with a new reply entry. In the thread, replies should be created with the current threads id, users id and message text.
    - Clicking the Add to Favorites should create a new entry in the favorites table with the thread's id and user's id.

2.1.3.3 *Commerce*

- **Offers**
    - On index page the application should retrieve all the offers paginated with the information.

- Clicking on an offer should redirect user and retrieve the offers information alongside its reviews from the database
    - Filling the review form with description and rating a new entry should be created in the reviews table with current offers id, user's id, description and rating.
    - Clicking delete review should remove the review entry from the reviews table.
    - Clicking on order button should display single order form
        - Filling the quantity and payment type and submitting the order should create a new order with business id and consumer's id in the orders table and also the order details with the newly added order id and payment method as well as an order offer entry with the amount and quantity.
    - Clicking on Edit Offer should take the user to the edit offer form.
        - Clicking Update Offer should update all the offers information in the offer table.
        - Clicking Remove Offer should remove the offer entry from the database.
- When the users click Search the offers that have checked categories, price lower than the minimum price should be retrieved paginated and displayed to the user.
- Clicking on Create New Offer should redirect user to create new offer form with a Preview Offer box.
    - Filling the form should display on the Preview Offer box how the final offer will look like.
    - Submitting the form should create a new entry in the offers table.
- By clicking the Add to Compare List the selected offer should be added to the compare list session as well as retrieve the necessary information needed to display the session entry in the compare list.
    - In case the offer is already in the Session nothing should be added.

        - Clicking on any of the Sorting Button option should sort the offers descending based on that sorting option
        - Clicking Clear Compare List should create the Compare Session from the offers.
    - Add to Favorites button should be yellow in case the user hasn't added the offer to favorites and red in case he has.
    - Clicking the Add to Favorites should create a new entry in the favorites table with the offer's id and users id.
        - In case the offer is already favorited the item should be removed from favorites.
    - Clicking on the Add to List item a new entry in the lists item table should be created with the list id, users id and offers id
        - In case the item is already in the list the item entry should be removed from favorites.
    - Clicking on the Add to Cart should create a new entry in the Session Cart
        - In case the offer is already in the cart nothing should happen.
        - The Cart entries should be grouped by the Offers user.
            - Filling the amounts and payment method and clicking Order a new order entry should be created in the orders table with the two users ids, a new order details should be created with the payment method, and for all the order offers corresponding entries should be created in the order offers with their ids and, as well as the newly made order id in both details and order offers tables.

- **Requests (Businesses)**
    - Index page should retrieve, paginate and display all the user's requests from the database showing it in list from with the consumers name, offer names, date of request and its status.
    - Clicking the Search button should take the user search input, checked statuses and payments and retrieve and display the corresponding requests.

- Clicking on the request should retrieve and display all the information's from the order, order details and order offers tables.
    - In case the status of request is not in Completed, Canceled or Closed the business editable information should be displayed in the format of inputs and a Update Order Button should be displayed.
        - Clicking the update button, the order details should be correspondingly updated.

- **Orders (Consumers)**
    - Index page should retrieve, paginate and display all the users' orders from the database showing it in list form with the businesses name, offer names, date of order and its status.
    - Clicking the Search button should take the user search input, checked statuses and payments and retrieve and display the corresponding orders.
    - Clicking on the order should retrieve and display all the information's from the order, order details and order offers tables
        - In case the status of order is in Ordered or Hold, the consumer editable information should be displayed in format of inputs and a Update Order Button should be displayed.
            - Clicking the update button, the order details and order offers information should be correspondingly updated.
    - In case the order is in Ordered or Hold, status a Cancel Order button should be shown
        - Clicking the button should update the status order table of the current order.

2.1.3.4 *Consumer Storage*

- **Lists**
    - Index page should retrieve all the users lists from the database and display in a list form.
        - Clicking on a list item should retrieve the corresponding list offers and display to user.

        - Clicking on Remove List Offer should remove the entry form the list offers.
        - Clicking on List Offer should redirect the user to the offers page.
    - Clicking on Remove List should remove the List and its Offers from the database.
- **Favorites**
    - Index page should display all the users' favorites.
        - Clicking on Remove Item from Favorites should remove the corresponding entry from the favorites table.
        -

2.1.4 Non-functional requirements

- The application should be easily expandable and maintainable.
    - This is ensured by using Laravel framework which uses design patterns and object-oriented principles.
- The application should keep passwords and any other private information which is confidential encrypted as well as hidden when displaying to the user.
    - This was done by manually encrypting the passwords and messages.
- The platforms performance needs to be efficient and fast as possible in order to ensure smooth and enjoyable experience.

2.1.5 Software Design Specifications

2.1.5.1 *Design Patterns*

The application is built using Laravel framework which uses the Model-View-Controller (MVC) Pattern. The MVC pattern splits the application and functionality into three groups Model, View and Controller. [29] [30] [31]

- **Model** is the connection and the data exchange with the database
- **View** is data representation logic.
- **Controller** is user action and input interpreter used to command model and view.

Building the application with MVC design pattern allows for functionality and organization of the code was done more efficient also gaining ability to be easily expanded in the future. This way the application has a library of the various functions as well as building blocks which can be implemented and or reused for future use. This allowed the development of the application to go faster and in more organized manner. Overall, using the Laravel Framework and their MVC provided better organization, structure and reusability of code.

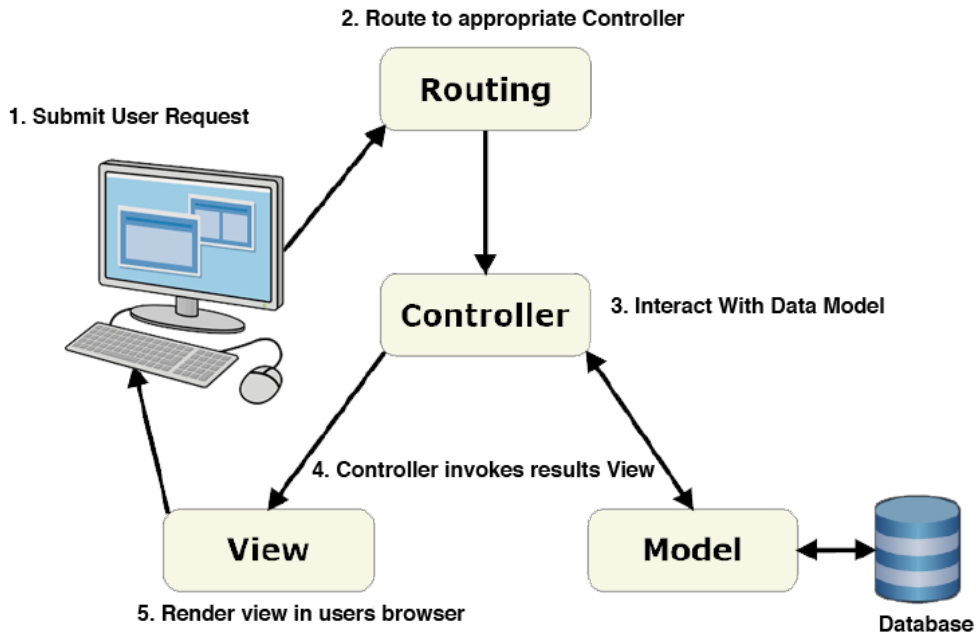


Diagram 1: *Example of MVC flowchart.*

2.1.5.2 *Database Design*

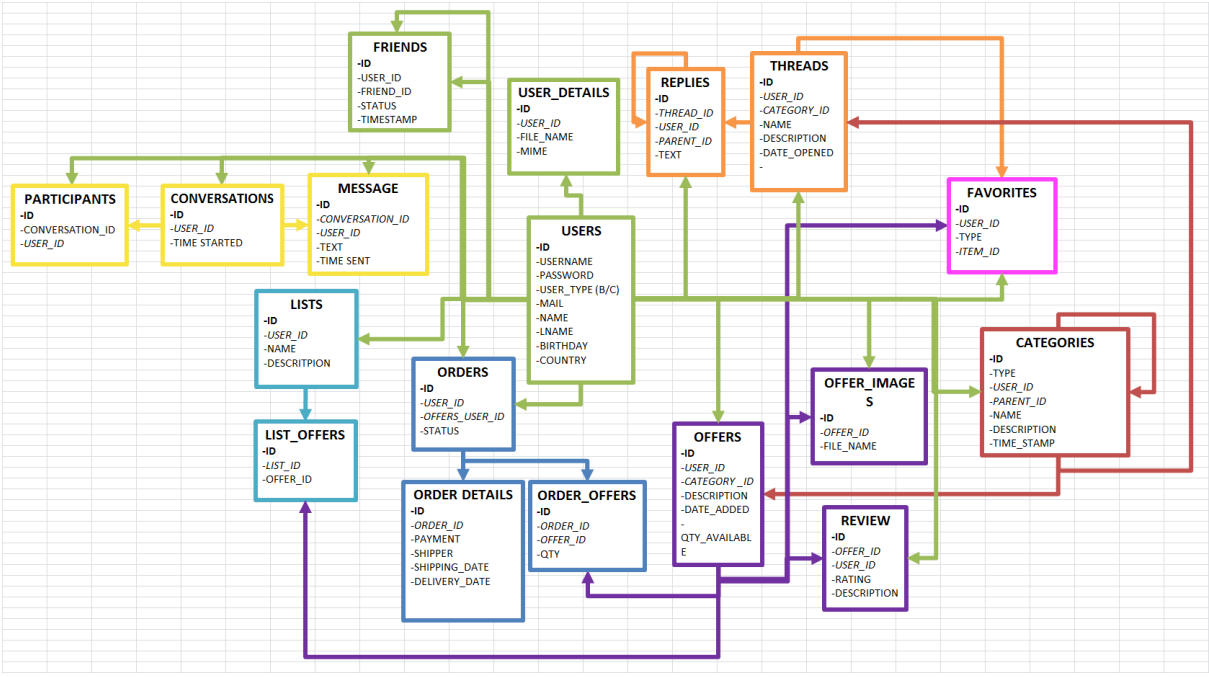


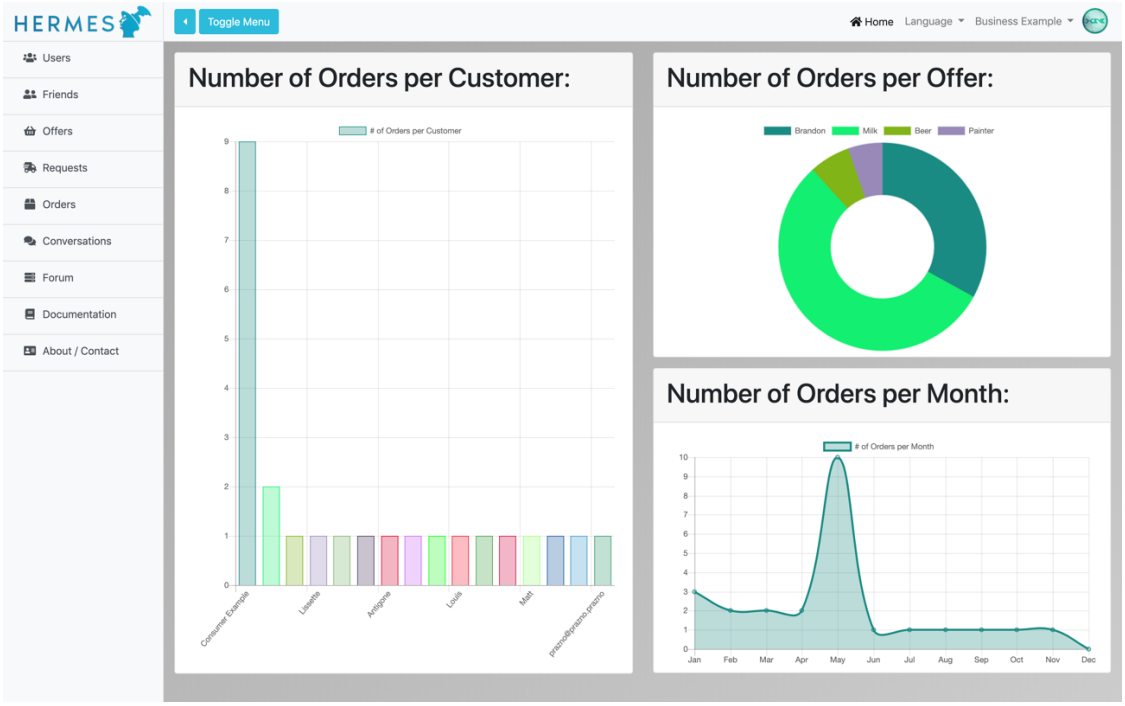
Figure 1. *The database design and all of its tables.*

Database communication is done using SQL which stands for Structured Query Language. It is the usual choice for developing relational databases. SQL statements are used to perform actions such as inserting, retrieving, updating etc. Database for the project consists of 18 tables which grouped by the feature units of the application are:

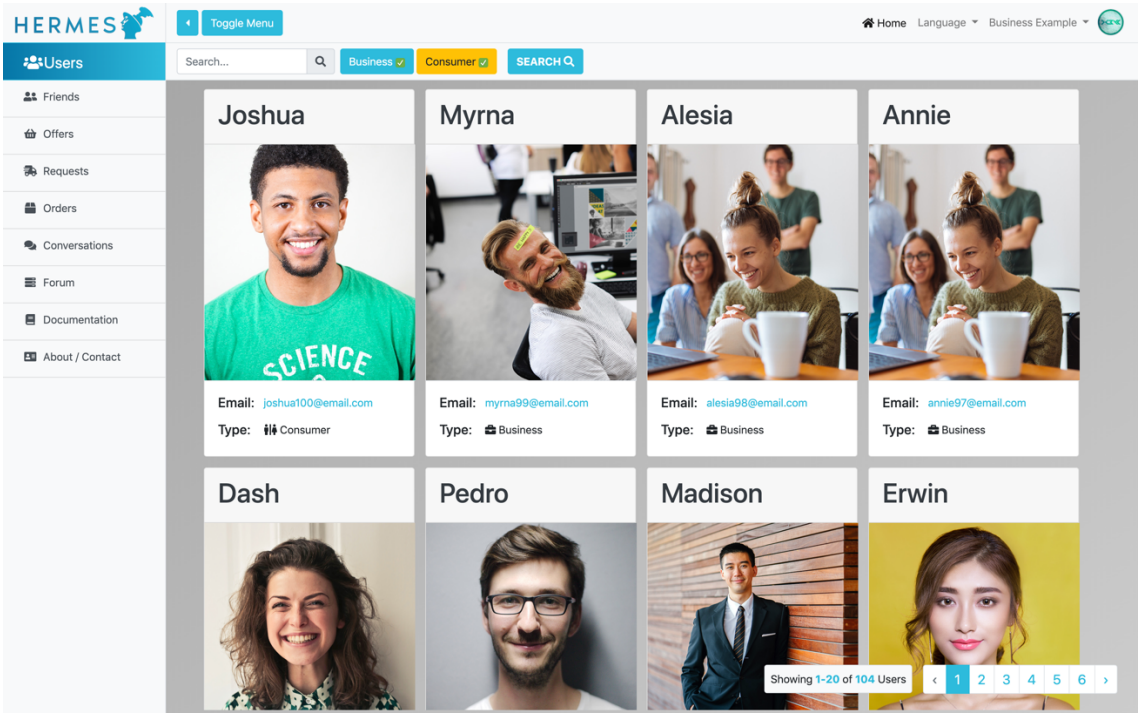
- User
  - Users, User Details
- Friends
  - Friends
- Messaging
  - Conversations, Participants, Messages
- Forum
  - Threads, Replies
- Offers
  - Offers, Offer Images, Reviews
- Orders/Requests
  - Orders, Order Details, Order Offers

- Lists
  - Lists, List Offers
- Categories (For Offers and Threads)
  - Categories
- Favorites (For Offers, Threads and Users)
  - Favorites

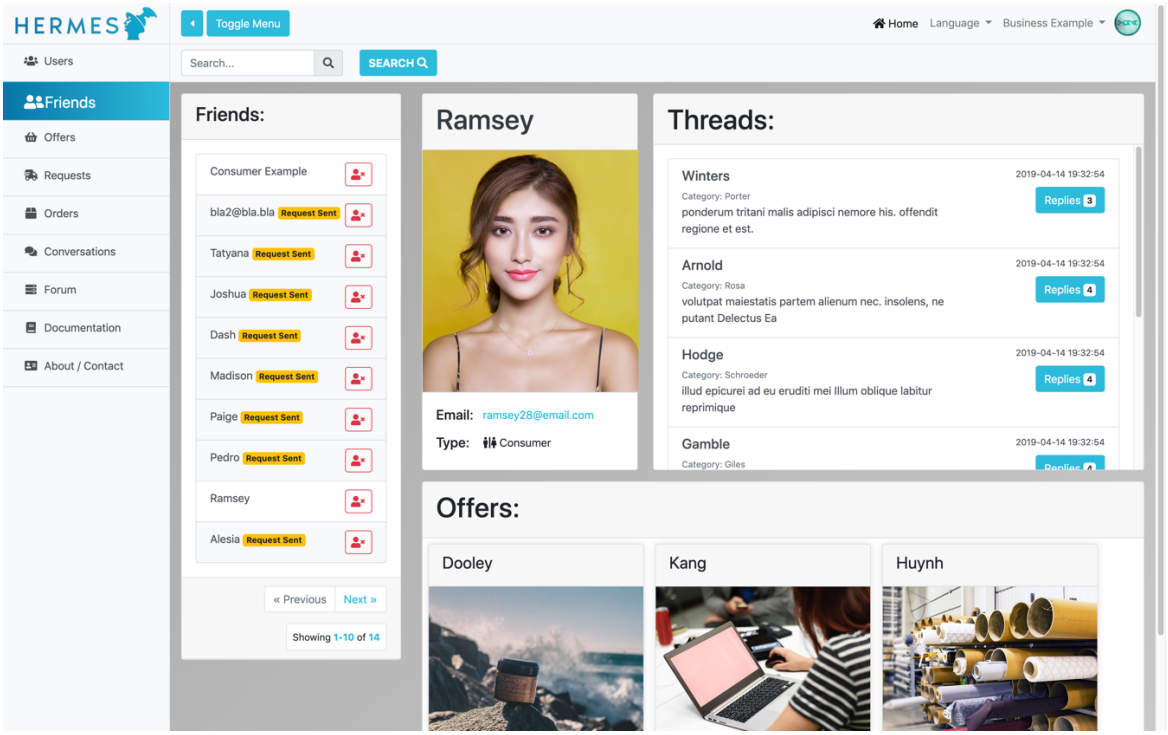
2.1.5.3 *GUI Design*



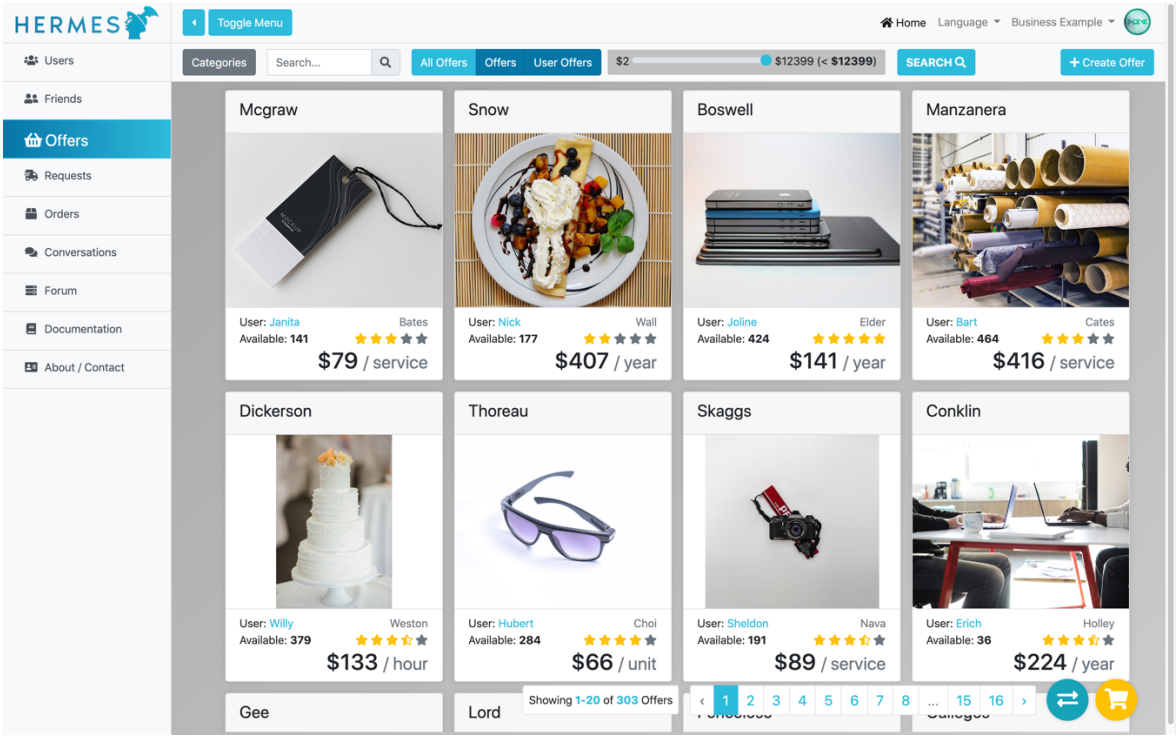
Picture 2. *Screenshot of Business Home Page.*



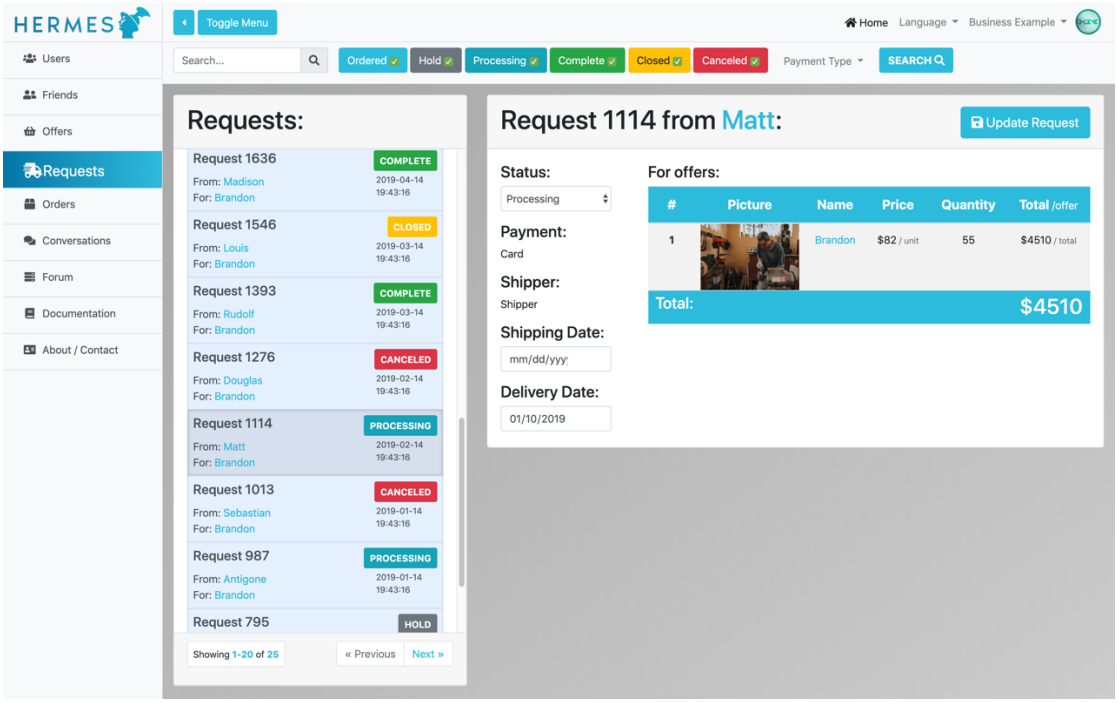
Picture 3. *Screenshot of Users Page.*



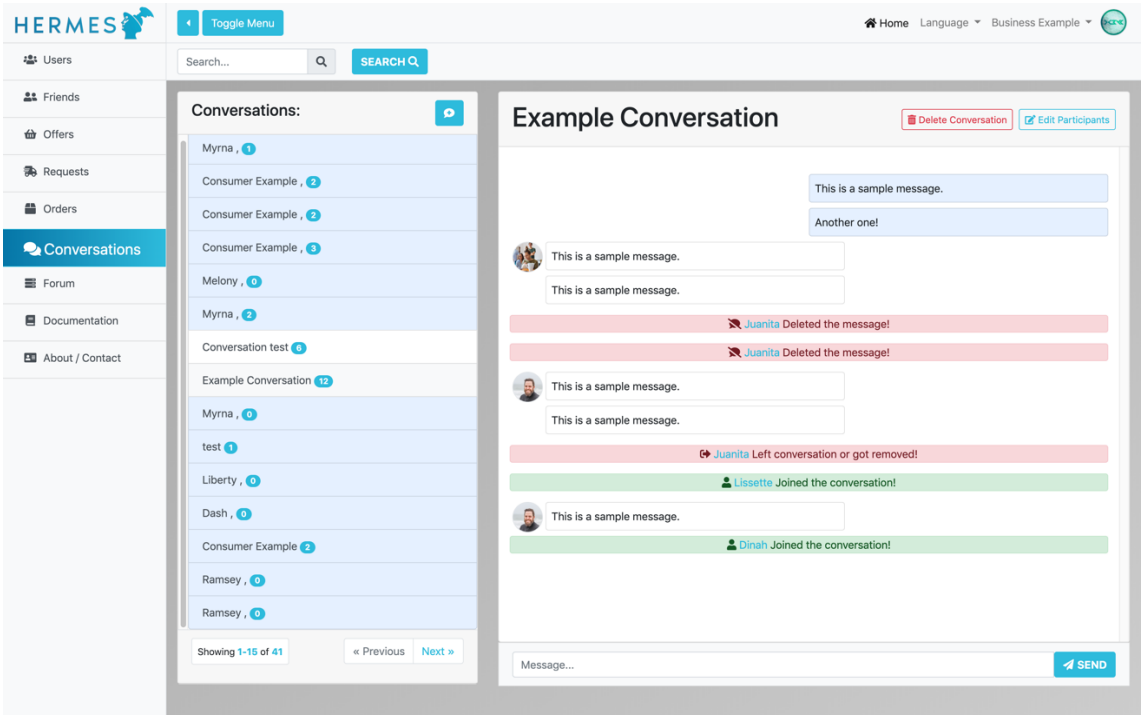
Picture 4. *Screenshot of Friends Page with friends' profile open.*



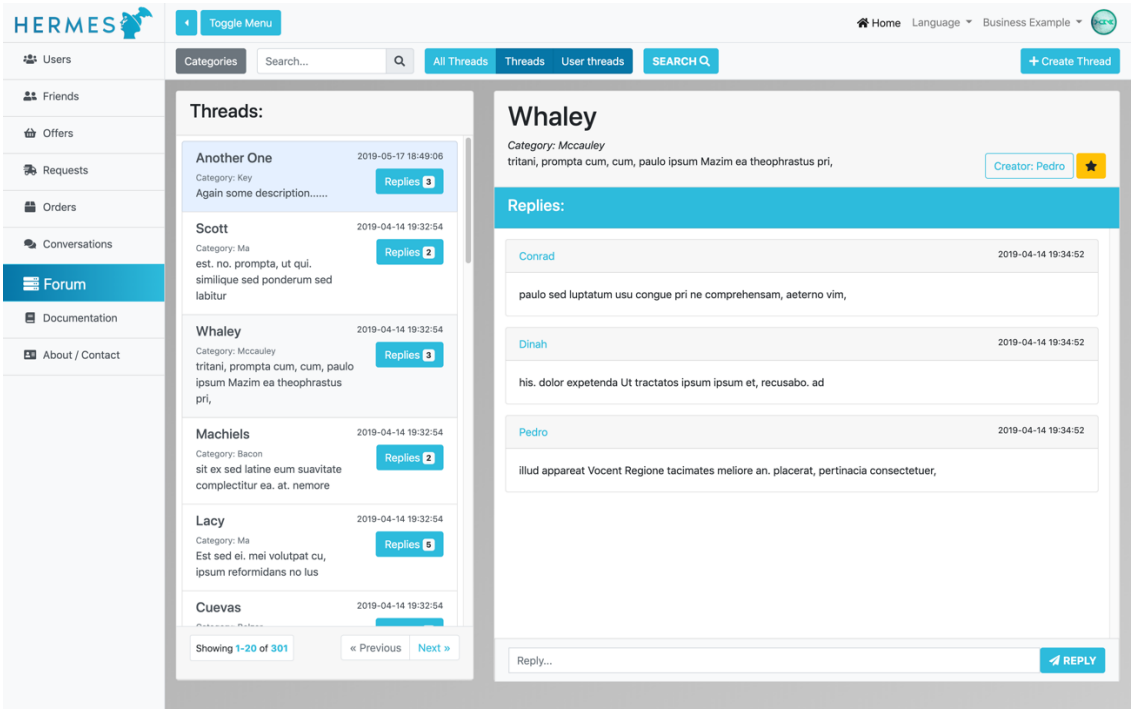
Picture 5. *Screenshot of Offers Page.*



Picture 6. *Screenshot of Requests Page.*



Picture 7. *Screenshot of Conversations Page.*



Picture 8. *Screenshot of Forum Page.*

2.2 List of design requirements

- **Convenient**
    - Simplicity of navigation and use.
- **Expandable**
    - General good coding practice.
    - Object orientation.
    - Laravel framework.
        - In order to ensure industry standards of organization as well as principles.
    - MVC design pattern.
    - Clear documentation.
- **Secure**
    - Encrypted and hidden information.
    - Checks who is the current logged user.
    - Validation of any user input in order to prevent SQL injection.

2.3 Development milestone

- Designing the database
- Organizing the MVC design pattern
- Learning the Laravel and Bootstrap Framework
- Register/Login system
- Offers
- Order/Cart system
- Comments and ratings option
- Searching, Filtering and Comparing options
- Forum
- Conversations
- Messaging system
- User Profiles
- Business pages, consumer profiles
- Friends
- Lists/Favorites
- Business statistics
- Consumer recommendations

2.4 Proposed timeline

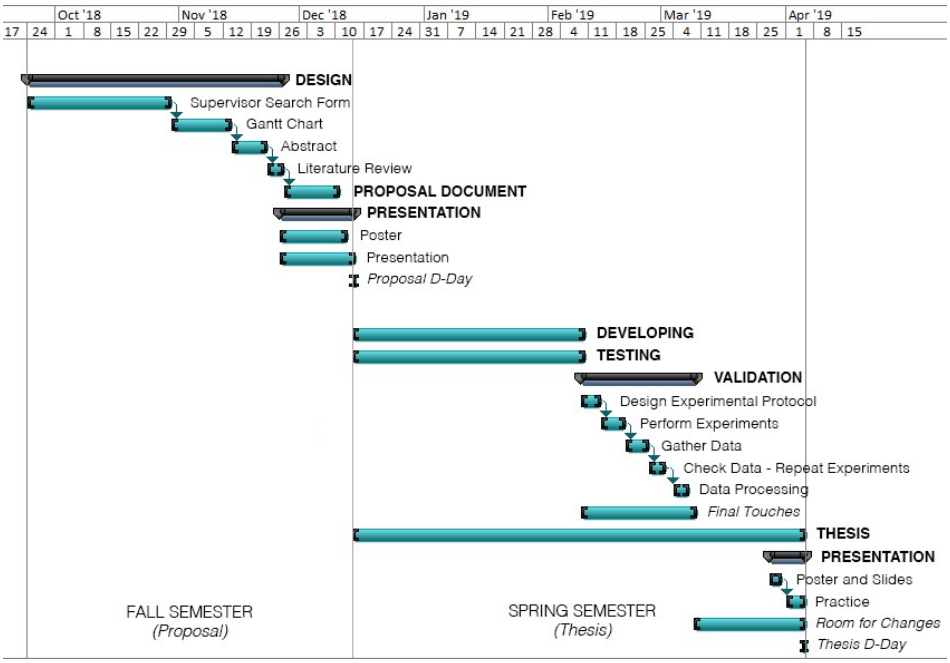


Figure 1: *Example of the GANTT chart*

Task Name	Duration	Start	Finish
Fall Semester (Proposal)			
DESIGN	**46 days**	**Mon 9/24/18**	**Mon 11/26/18**
Supervisor Search Form	26 days	Mon 9/24/18	Mon 10/29/18
Gantt Chart	11 days	Tue 10/30/18	Tue 11/13/18
Proposal Abstract	7 days	Wed 11/14/18	Thu 11/22/18
Literatire Review	2 days	Fri 11/23/18	Mon 11/26/18
PROPOSAL DOCUMENT	**10 days**	**Tue 11/27/18**	**Mon 12/10/18**
PRESENTATION	**15 days**	**Mon 11/26/18**	**Fri 12/14/18**
Poster	13 days	Mon 11/26/18	Wed 12/12/18
Presentation	15 days	Mon 11/26/18	Fri 12/14/18
Proposal D-Day	***1 day***	***Fri 12/14/18***	***Fri 12/14/18***
Spring Semesters (Thesis)			
DEVELOPMENT	**42 days**	**Fri 12/14/18**	**Sat 2/9/19**
TESTING	**42 days**	**Fri 12/14/18**	**Sat 2/9/19**
VALIDATION EXPERIMENT	**22 days**	**Sat 2/9/19**	**Sat 3/9/19**
Design Experimental Protocol	4 days	Sat 2/9/19	Wed 2/13/19
Perform Experiments	4 days	Thu 2/14/19	Tue 2/19/19
Gather Data	4 days	Wed 2/20/19	Mon 2/25/19
Check Data - Repeat Experiments	4 days	Tue 2/26/19	Fri 3/1/19
Data Processing	4 days	Mon 3/4/19	Thu 3/7/19
Application Final Touches	*22 days*	*Sat 2/9/19*	*Sat 3/9/19*
THESIS DOCUMENT	**81 days**	**Fri 12/14/18**	**Fri 4/5/19**
PRESENTATION 2	**7 days**	**Thu 3/28/19**	**Fri 4/5/19**
Poster and Slides Preparation	3 days	Thu 3/28/19	Sat 3/30/19
Practice for Presentation	5 days	Mon 4/1/19	Fri 4/5/19
Room for Possible Changes	**21 days**	**Sat 3/9/19**	**Fri 4/5/19**
Thesis D-Day	***1 day***	***Fri 4/5/19***	***Fri 4/5/19***

Table 1: *Example of the GANTT chart table with start, finish and duration of each task*

2.5 Contingency planning

Most of the contingency planning requires not delivering everything that was planned and or having to minimize some functionalities. This was mainly due to the time constraint and the manpower available due to having a one-man army team. The task which took the most time was learning how the Laravel and Bootstrap frameworks work. However, they are there to provide more time in the case one successfully learns how to wield them. Some of those contingencies are:

- Not providing full functionalities of the applications features
    - o *Solution* to this would is to reduce the intended complexity level of the application and provide at least the core functionalities of each proposed option in order to have as functioning application which can come closer a real-world application. Reducing features functionality such as having:
        - Lower options of filtering.
        - Weaker recommender system.
        - Lower amount of page personalization options.
        - Not providing all aspects of security.

- Not making the application as cross-platform as possible and not making personalization as expected and or not as versatile.
    - Due to vast amounts of display resolutions.
    - Due to user providing pictures.
        - Pictures being distorted, skewed, small etc.
    - Functionality messed up due to different web browsers and sizes.
    - Not as mobile friendly.

    o *Solution* is to use a front-end framework which speeds up the development significantly and provides tools to provide the same

experience on most if not all platforms. Also, this way from beginning the focus is to make application work as intended without having to worry as much for the visual part.

- Not having the functions, queries and options as efficient as they could be.
    - o *Solution* is to have a bottom line of application efficiency in order to provide a fluent enough experience to satisfy the user.

3 Description of hardware

The users would need internet connection and a web browser in order to access the application from either PC or a mobile phone. Alongside with skills such as basic computer literacy in order to properly use it.

The development of the application was done on:

- ***Laptop***: MacBook Pro (Retina, 15-inch, Mid 2014)
- ***Processor***: 2.2 GHz Intel Core i7
- ***Memory***: 16 GB 1600 MHz DDR3
- ***Graphics Card***: Intel Iris Pro 1536 MB

These specification and even lower ones should be enough to serve the user as the development has both the server and user parts running on the same machine. While in the scenario where the application is deployed the real world, the user would have faster performance not taking into account the internet speed and the access to the servers.

4 Description of software

The application was built using a framework called Laravel. Laravel is free, open-source cross platform PHP web framework based on Symfony, it is used for developing web applications using a Model-View-Controller design pattern. [32]
Operating system of choice for developing the web application is Mac OSX.
Text editor of choice to build the web application is Sublime Text 3 which provides feature called Package Control which allows the addition of plugins which increase the speed of writing code.

Picture 9. *Logo of the Laravel Framework.*

Regarding the languages the web page is written in HTML, stylized using CSS and jQuery and made run with PHP and JavaScript. The languages, technologies and libraries used to develop the application are:

- **HTML / CSS**
  - Creation and stylization of the web page.
  - **Bootstrap**
    - Framework for the base styling of the page.
- **SQL**
  - Communication and development of database.
- **PHP**
  - Actual logic and functionality of the web page.
  - **Laravel**
    - PHP web framework.
- **JavaScript**
  - **jQuery**
    - User and front-end logic and stylization of the page.

4.1 Laravel

Some of the main tools used to and for the building of the application from the Laravel framework are Artisan, Laravel Schemas and Migrations, Eloquent ORM, Blade and (security) CSRF protection etc.

4.1.1 Artisan

Laravel comes with a built-in command-line which provides useful commands that assist the developer during the development of the application. The command-line is called Artisan, it creates and handles Laravel Project environment. Artisan gives ability to perform repetitive and tedious programming task which normally are to be done manually. Tasks such as managing database, creating skeletons(templates) for any of the models, views, controllers etc. It increases the speed of writing code. [32]

4.1.2 Schema, Migrations

Laravel also comes with tools for managing a database specifically Schemas and Migrations. Schemas gives ability to agnostically manipulate tables and Migrations are like version controls for the database. Migrations are usually contacted with a Schema Builder which allows modification of database tables while recording all the versions. Overall, it gives ability to easily create, modify and keep track by go through version using rolling back feature without manually writing SQL statements. [32]

4.1.3 Eloquent ORM

Besides the ability of managing the database on its own Laravel provides Eloquent ORM, which gives ActiveRecord which is the Model in the MVC design pattern. Model is connected with a corresponding table which gives Object Relational Mapping system. This creates objects which have persistent storage in the database making it easy to get and control the information. This allows the connection and data extraction easy, having the ability to easily make relationships and jump from one to another as well as have access to all various SQL commands which the Eloquent provides makes creating complex queries easy. [32] [33]

4.1.4 Blade

Laravel also comes with a tool for managing the front-end of the application called Blade. Blade is a templating engine which brings object-orientation philosophy to the front-end. Contrary to other templating engines Blade doesn’t restrict from using plain PHP code, however with the features which Blade comes with this is rarely done. [32]

It allows the creation of easily readable and fast to implement templates. Which keeps everything organized in its own file, making every View like a module that can be reused anywhere in the code as long as the correct Model is sent to it.

4.1.5 Routes and Controllers

Laravel Routing System makes the creation of URL routes that connect a user to the correct Controller and its targeted function, fast and easy. Combined with artisan and previous feature in several lines of code one can create a base MVC pattern which is connected to the database and the application itself. [32]

4.1.6 Security

Laravel also comes with tools that provide security out of the box which keep the user data secure. Feature such as Authentication system, CSRF and SQL Injection protection, Cookie encryption etc. [32] [34]

4.1.6.1 *Authentication system*

Laravel also provides authentication system with two authorizing tools, gates and policies. Gates use Closure programming philosophy which check if the user is authorized to perform an action, and policies connect this to a model or resource. This allows user registration, logging in and usage of the application secure with up to date security standards. [32] [34]

4.1.6.2 *CSRF Protection*

Laravel gives protection to cross-site request forgery (CSRF) attacks, where the authenticated users commands are performed by an unauthorized user. The way Laravel secures the application from these attacks its by using CSRF “tokens” which are generated for every user. This token is then used to check whether the user performing the actions is in fact the authorized user. [32] [34]

4.1.6.3 *SQL Injection Prevention*

Structured Query Language (SQL) injection is one of the most common web hacking techniques. It is an attack which attempts to trick the database into executing malicious code in order to retrieve, add, modify, and or delete records in the database. SQL injection usually occurs when user is asked from some input, like a

username, and instead of a name an SQL statement is given which is unknowingly ran database. [34] [35]

Laravel protects itself by using prepared statements which escapes any such malicious user input. Prepared statements send the query and the data separate cutting the ability of injecting SQL statements. Even if the user adds a new input with a quote sign to be able to inject SQL statement this is escaped by Eloquent and invalid statement will be reported to database. [32] [34] [36]

4.1.6.4 *Cookie Encryption*

Laravel also protects the cookies from being altered or obtained by other users. This is done by using a random unique generated Application key or Encryption key. The key is then used for encrypting and hashing the cookies. The encryption prevents unauthorized user from seeing it and hashing for checking if any character was changed in the transaction. [32] [34] [36]

Also, on the entrance of every Controller function where only one specific user should be able to access, his identity is checked. In case someone tries to access something which they are not authorized they are redirected to a 403-error page. Which sates to the user that they don't have access to this particular page. This provides a Gate which protects any user's information from being modified by external users which accesses the page by writing their own URL's.

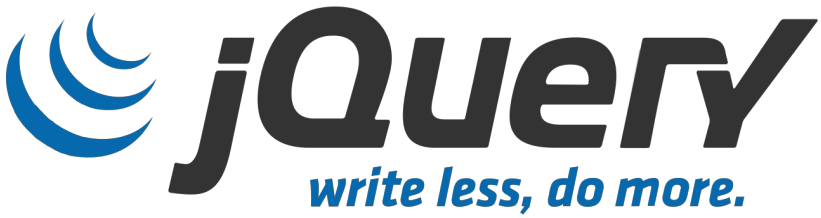
4.2 Bootstrap

Bootstrap is an open-source framework which allows to quickly build an entire front-end of an application. It uses HTML, CSS, JS and jQuery as its background languages. It significantly cuts down the development time of the web page styling by providing prebuilt components, responsive grid system which is aimed at also mobile web pages, and other plugins from jQuery. Overall, Bootstrap provides a powerful library of components which cut down development of the front-end. [37]

***Picture 10**. Logo of Bootstrap framework.*

4.3 jQuery

jQuery is widely used JavaScript framework which was used for front-end and deploying AJAX in the application. jQuery provides a simpler way of traversing a HTML DOM tree than JavaScript as well as plethora of different kinds of function. This way grabbing, modifying and using any element or class in the page is easy. Minimizing the amount of time, one needs to spend on the development of the front-end. [38]



***Picture 11**. Logo of jQuery library.*

.

5 Planned validation experiments

Validation of the platform is conducted to test both the users experience, the actual performance of the application, the object orientation and expendability.

User and programmer experiences are validated using forms, polls and questionnaires to check the user's satisfaction regarding the simplicity, security, ease of use and expandability.

The application performance is validated using RAIL Model, Load and Stress testing. RAIL Model testing is performed to measure responsiveness of the application using Chrome Developer Tools. [39] Load testing is where different database simulations of varying amounts of data are performed, where the query efficiency is tested by measuring execution time. Stress testing is conducted by imitating server connection and query overload up until it crashes in order to test the full capabilities of the system. Overall, the application validation will totally test the application both on the frontend meaning users experience and backend as the performance of the application.

The object orientation and expendability are validated using source code metrics on the application PHP code, using a plugin called PHPMetrics.

In conclusion having an application with slow response time and weak performance can produce big issues.

5.1 Load Testing

Load Testing is performed to measure and discover any performance issues prior to its deployment to the users. Load Testing simulates scenarios of various sizes of user's activities trying to mimic real-world in order to predict where it can go wrong or where it immediately needs to be fixed. This way the performance impact can be quantified by measuring various response time. [40]

5.2 Stress Testing

Stress Testing, also known as Fatigue Testing, is used to find the breaking point of the application and system. This done by providing the application and system with

too many tasks which make it stop working at some point. The name of this moment is called breakpoint of the database system. [41]

5.3 RAIL Model

RAIL Model stands for Response, Animation, Idle and Load which are the aspects which it measures/checks. The RAIL Model can be tested using Chrome DevTools which come with Google Chrome. It is cantered around the user and how his experience is by breaking it into key actions. The users in various ways see performance delays relying on their hardware specification of network connection. [39]

- **Response**
  - Goal is to have response under 50ms in order not to break the link between action and reaction.
- **Animation**
  - Goal is to produce an animation under 10ms which aims at fluidity as the user are aware when the frame rate varies.
- **Idle**
  - Goal is to lower inactive or idle time in order to response to user in under 50ms.
- **Load**
  - Goal is to lower load time as the user attention is lost with a thought that the task is maybe even broken. Sites that have lower load time usually have longer user activities.

Picture 12. *The four aspects of RAIL Model.*

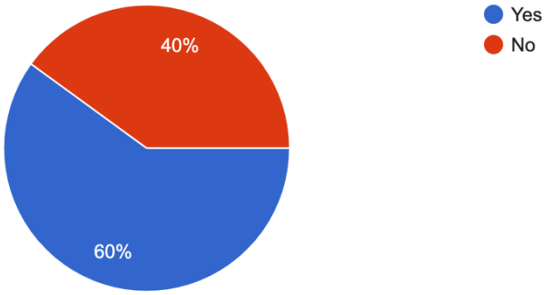
5.4 Results

5.4.1 Survey Results

5.4.1.1 *General Questions*

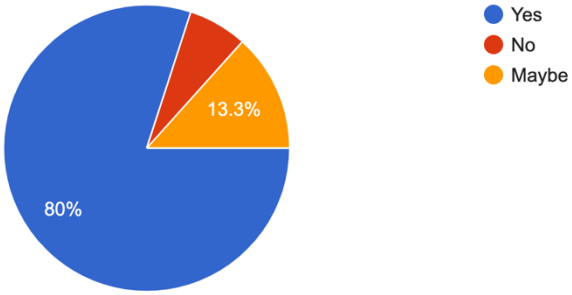
Do you have any experience using Social-Commerce platforms?

15 responses



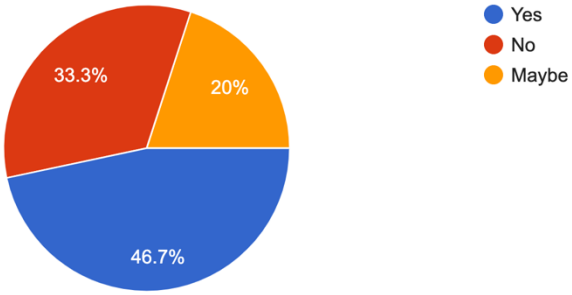
Did you find the experience fluent and snappy?

15 responses



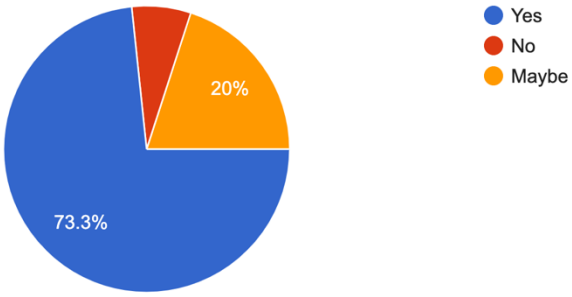
Do you find the application invokes security?

15 responses



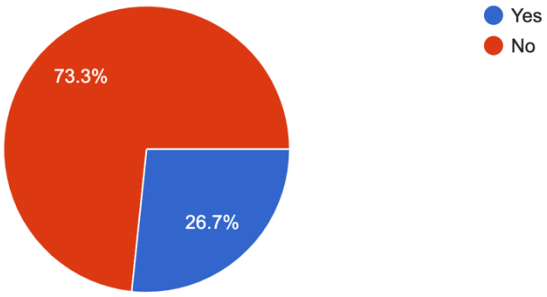
Do you feel the platform is easily expandable?

15 responses



Are you a programmer?

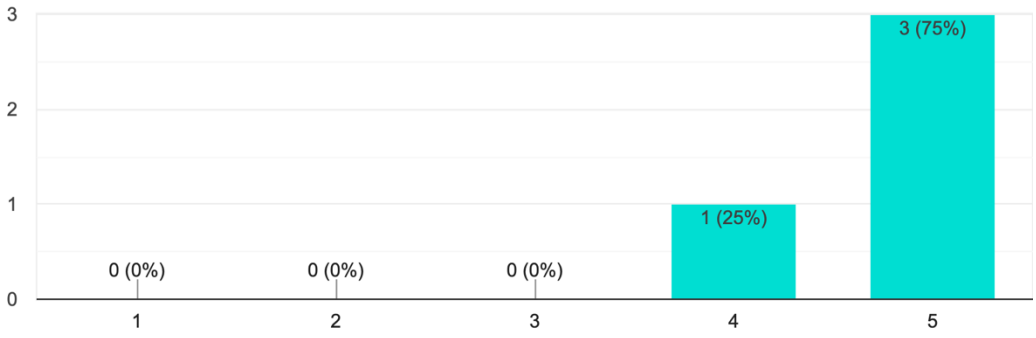
15 responses



5.4.1.2 *Programmer Questions*

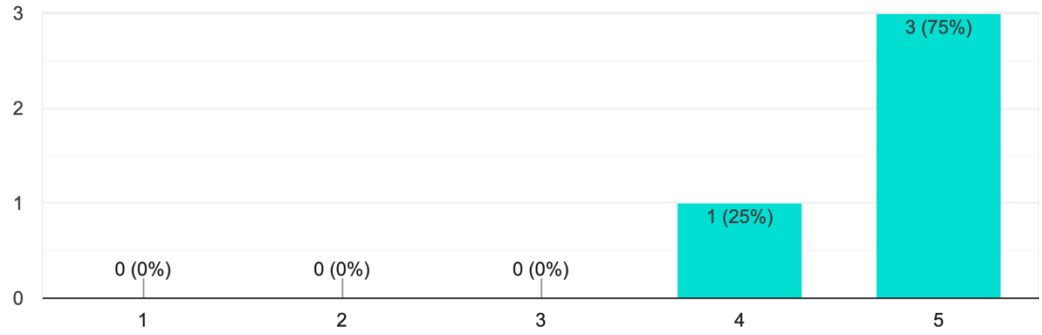
Rank the following of object-oriented principles?

4 responses



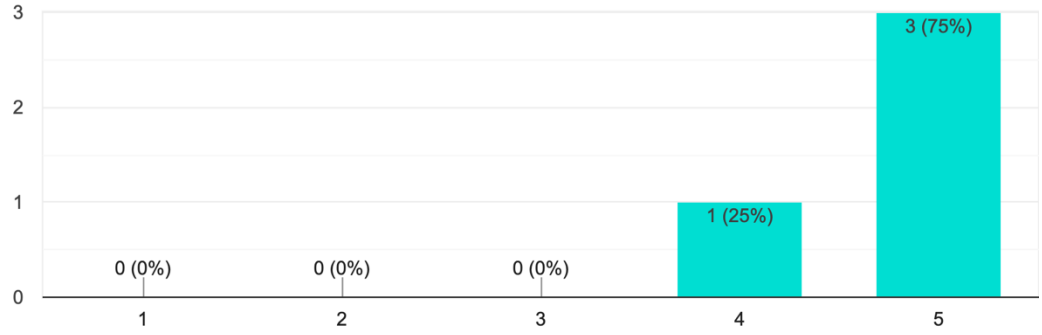
Rank the database design?

4 responses



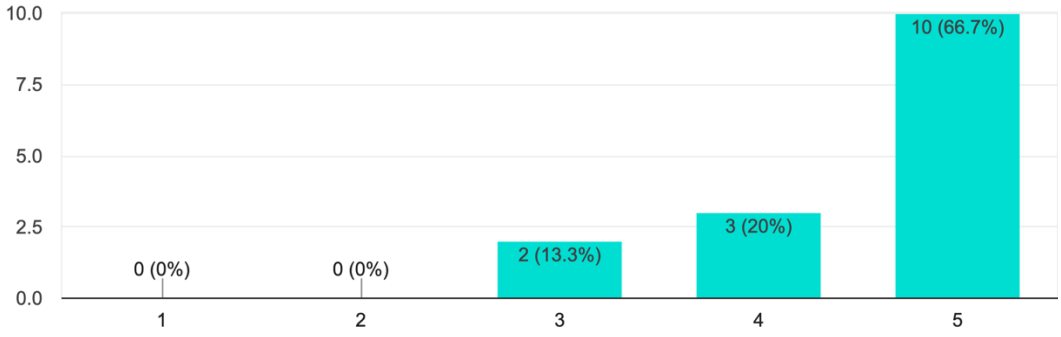
Rank the security of the application?

4 responses



How would you rank the convenience and ease of use of the platform?

15 responses



5.4.2 Stress and Load Tests Results

Stress and Load testing was conducted using Siege and Apache Benchmarking tool.

5.4.2.1 *Siege*

Siege	
Transactions:	4841 hits
Availability:	100.00%
Elapsed time:	59.16 secs
Data transferred:	184.81 MB
Response time:	0.30 secs
Transaction rate:	81.83 trans/sec
Throughput:	3.12 MB/sec
Concurrency:	24.67
Successful transactions:	4841
Failed transactions:	0
Longest transaction:	2.08
Shortest transaction:	0

5.4.2.2 *Apache*

Apache AB	
Server Hostname:	127.0.0.1
Server Port:	8000
Document Path:	/
Document Length:	4670 bytes
Concurrency Level:	5
Time taken for tests:	7.491 seconds
Complete requests:	100
Failed requests:	0
Total transferred:	562486 bytes
HTML transferred:	467000 bytes
Requests per second:	13.35 [#/sec] (mean)
Time per request:	374.553 [ms] (mean)
Time per request:	74.911 [ms] (mean, across all concurrent requests)
Transfer rate:	73.33 [Kbytes/sec] received

Overall, the results from Stress and Load testing are relative to the machine that is serving the application. Which is in this case the laptop from the hardware section where the app is developed. This way results provide less meaningful metrics to the application but more to the machine which is not a proper server to serve the application online despite the good results.

5.4.3 RAIL Model / Lighthouse (Google Chrome Audits) Results

The RAIL Model was tested using Lighthouse which is built in Google Chrome Developer tools under Audits section. The validation was tested without and with throttling. Throttling was done with Simulated Fast 3G and 4x CPU Slowdown in order to have an idea of a regular user with lower processing power of accessing the application.

Performance reflects the time needed to get the page fully loaded, accessibility refers to the ability of making the application usable to many users, best practices is the use of up to date solutions, SEO or Search Engine optimization refers to metadata present for search engines to use for their results.

ROUTE	PERFORMANCE (Throttling)	PERFOMANCE (No Throttling)	ACCESABILITY	BEST PRACTICES	SEO
/	100	100	72	86	100
about/contact	94	99	83	93	100
conversations	87	100	47	93	100
conversations/{conversation}	85	100	42	93	100
documentation	97	100	63	93	100
favorites	79	100	69	93	100
forum	84	100	58	86	100
forum/{forum}	81	100	47	86	100
friends	85	100	53	93	100
home *business	85	99	53	93	100
home *consumer	73	94	70	64	100
lists	92	100	65	93	100
login	92	100	81	93	100
offers	77	97	66	86	100
offers/create	89	100	67	86	100
offers/{offer}	87	100	67	79	100
offers/{offer}/edit	92	100	67	86	100
orders	85	100	63	93	100
orders/{order}	84	100	63	93	100
register	93	100	68	93	100
requests	85	100	63	93	100
requests/{request}	85	100	63	93	100
users	82	99	63	86	100
users/{user}	88	100	74	86	100
users/{user}/edit	87	100	67	86	100
AVERAGE	**86.72**	**99.52**	**63.76**	**88.76**	**100**

Table 2. Results from the RAIL Model Testing

Overall, the results from the RAIL Model are good. The performance is in the upper percentile of the average web page with throttling and without in the upper percentile of all webpages. Accessibility grade is low this is due to the use and combination of Bootstrap and jQuery which made violate some rules which lower the grade. Best practices are also good with some minor changes which can be made to get a full grade, and SEO with full grade of 100.

5.4.4 Software Metrics (PHPMetrics)

LOC	
Lines of code	2923
Logical lines of code	2171
Comment lines of code	753
Average volume	213.32
Average comment weight	27.69
Average intelligent content	27.69
Logical lines of code by class	28
Logical lines of code by method	9

Object oriented programming	
Classes	78
Interface	0
Methods	232
Methods by class	2.97
Lack of cohesion of methods	2.27

Coupling	
Average afferent coupling	0.85
Average efferent coupling	2.1
Average instability	0.84
Depth of Inheritance Tree	2

Package	
Packages	9
Average classes per package	8.67
Average distance	0.13
Average incoming class dependencies	2.78
Average outgoing class dependencies	6.22
Average incoming package dependencies	0.67
Average outgoing package dependencies	3.11

Complexity	
Average Cyclomatic complexity by class	2.53
Average Weighted method count by class	4.5
Average Relative system complexity	86.5
Average Difficulty	2.79

Bugs	
Average bugs by class	0.07
Average defects by class (Kan)	0.3

Violations	
Critical	0
Error	5
Warning	9
Information	0

***Tables 3**. Results from PHPMetrics regarding the average values.*

Overall, on average the results are good regarding the expendability and object orientation. However, the results on individual basis indicate that the complexity and maintainability of controllers is high. This is expectable as that's where most of the application logic is located. This means it could use some refactoring in order to improve the results.

6 A look into the future

Technology usually goes in the direction of mimicking something from the nature or making it do something better which analogue way of doing it permits. [42] The applications of the future will probably try to mimic the traditional way of shopping as much as possible without its limitations and difficulties which it brings. For example, with the future of virtual reality e-commerce could be conducted in such form, where the user would be able to go to a virtual version of a store and walk around, try and buy whatever products they want.

6.1 Advances in the technologies relevant to the project

With the development of technologies such as Machine Learning and Artificial Intelligence, the algorithms for molding the platform to user experience will progressively improve. [19] [43] There will be progression from the recommender system to custom page design for each user. [44] The more we learn how to use the information we produce, the more we can create a personalized experience for the user. Also, with Internet of Things where internet access will be readily available to all devices in the future, the number of advances and possible applications is endless. [22].

6.2 Future improvements (on this project)

The application can be endlessly improved by:

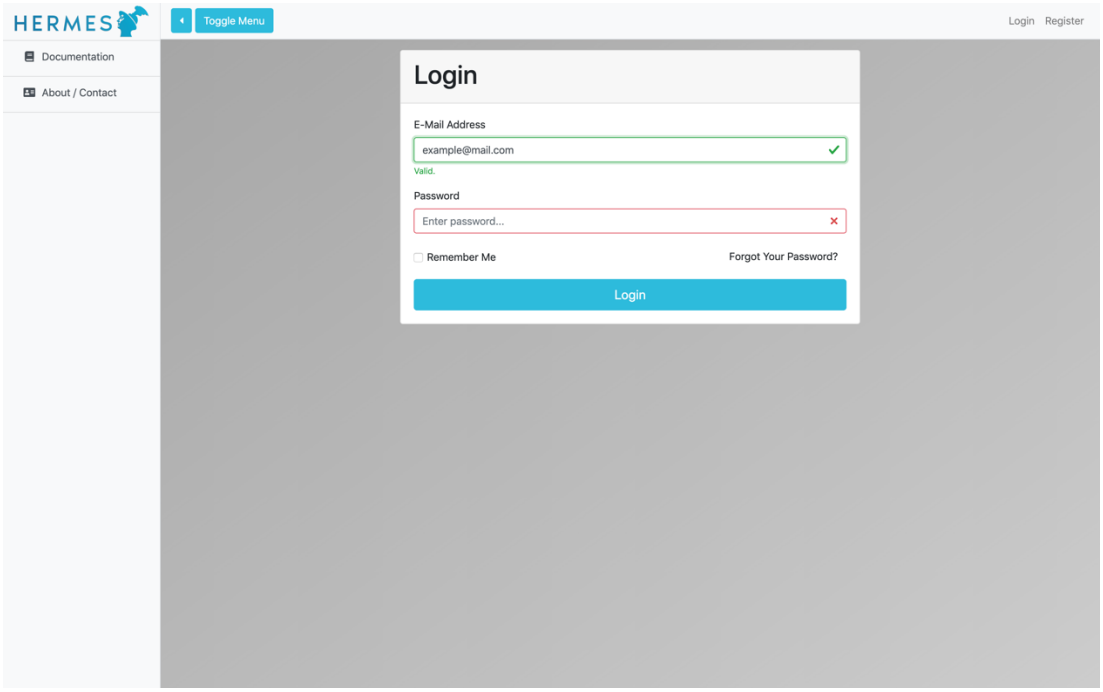
- Expanding the information which the database holds which would allow more features to be developed.
- Providing more options of filtering which will use the previously expanded information.
- More options of page personalization, such as having more options for the style of the profile rather than having a single one for everyone.
- Better algorithm regarding business statistics and consumer recommendations.
- Improve the efficiency of the used queries.

- More organized database which would better accommodate huge amounts of data.
- More notifications such as when a user gets or sends an offer, gets a message outside the given chat message etc.
- Admin panel where certain aspects can be modified easier by having it built as a feature rather than to code it. This way even people who may not know how to code will be able to make certain modifications to the page.
- Ability for Business to export a report of their statistics would be beneficial and provide a functionality which is not as common.
- Having a feedback on the user actions in order to have a greater feeling of interaction with the application.
- Ability for user to put labels on their offers such as Sold-out, Fresh, Sale etc.
- Ability to add tags on an offer which would help with searching and better categorizing.
- Ability for user to adjust positions of the pictures they upload, such as for an offer.
- Ability to choose thumbnail and or main picture for the offers.
- Ability to have different options on a single offer, such as to choose for example a color of an item.
- Introduce a gift card system.
- Introduce more pop-up hints as well as a tutorial of sort when first time logging in after registering etc.

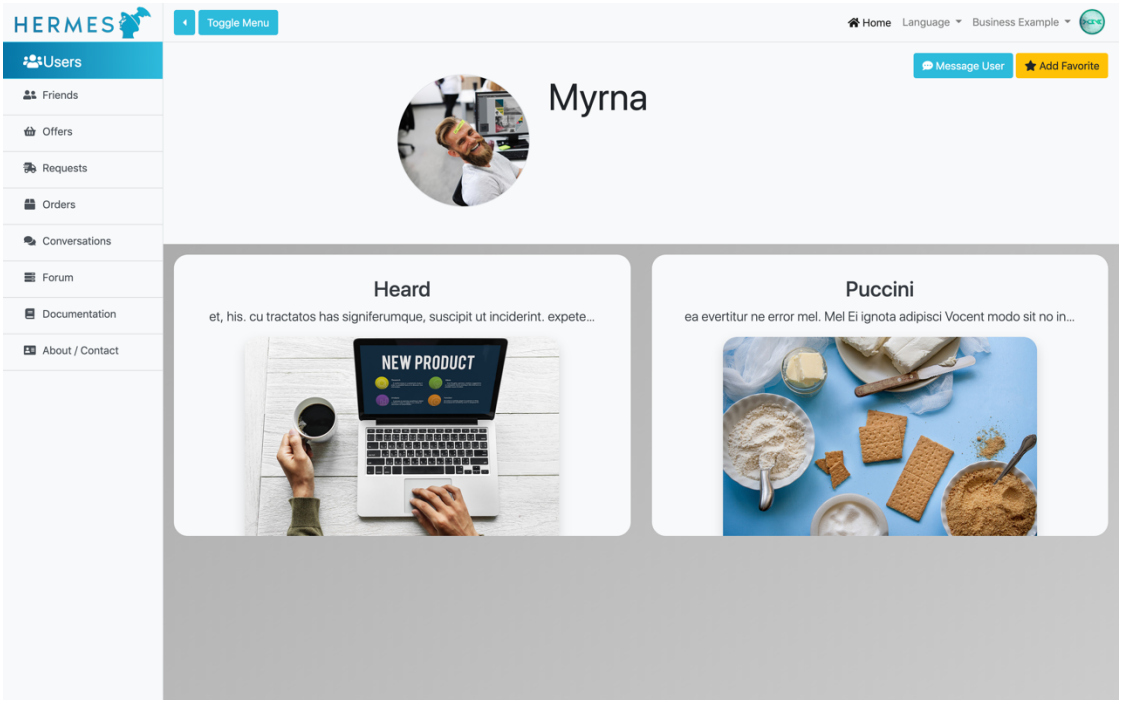
6.3 Insights into future applications

Social-commerce is the future of e-commerce where the social experience which is present in traditional way of buying goods is going to be made possible in the technological form. Where consumers will have their own local businesses connected, each communicating with each other in different shapes and forms creating a community dedicated to trading.

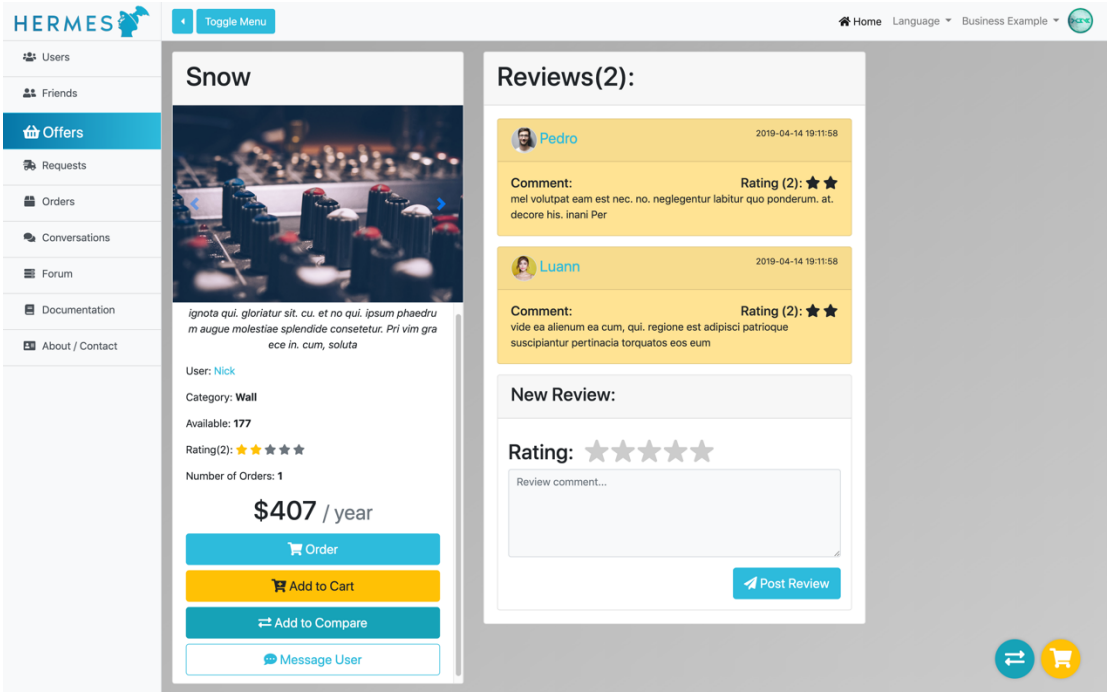
Appendices



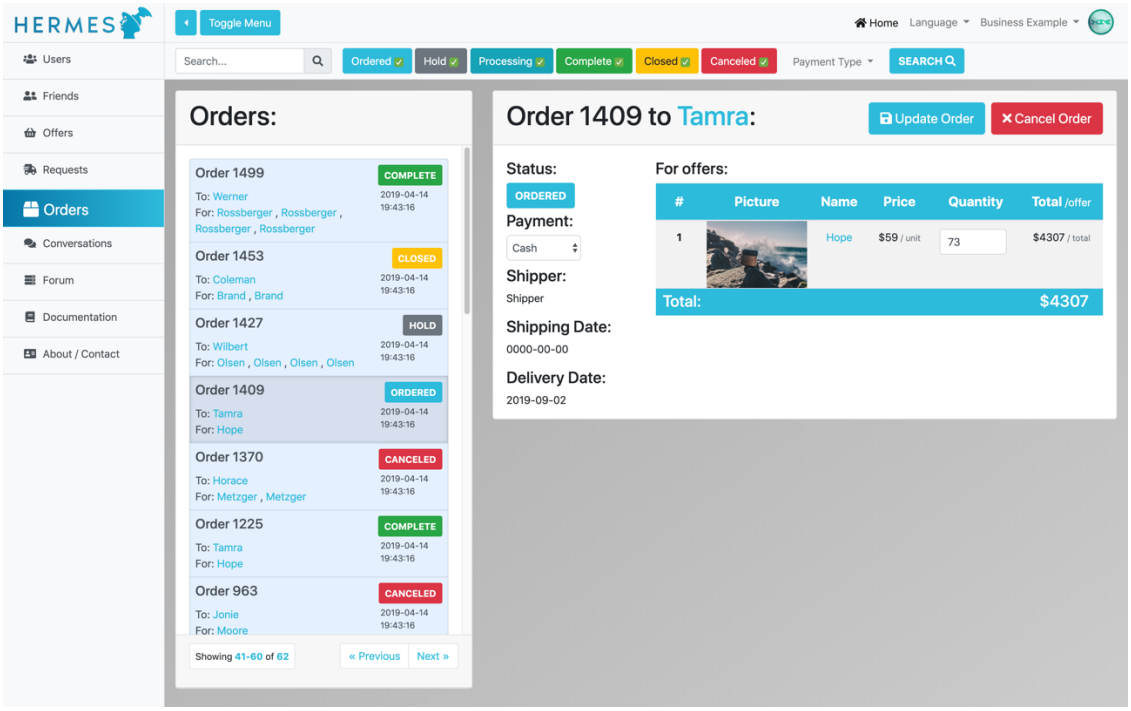
***Picture 13**. Screenshot of Login Page.*



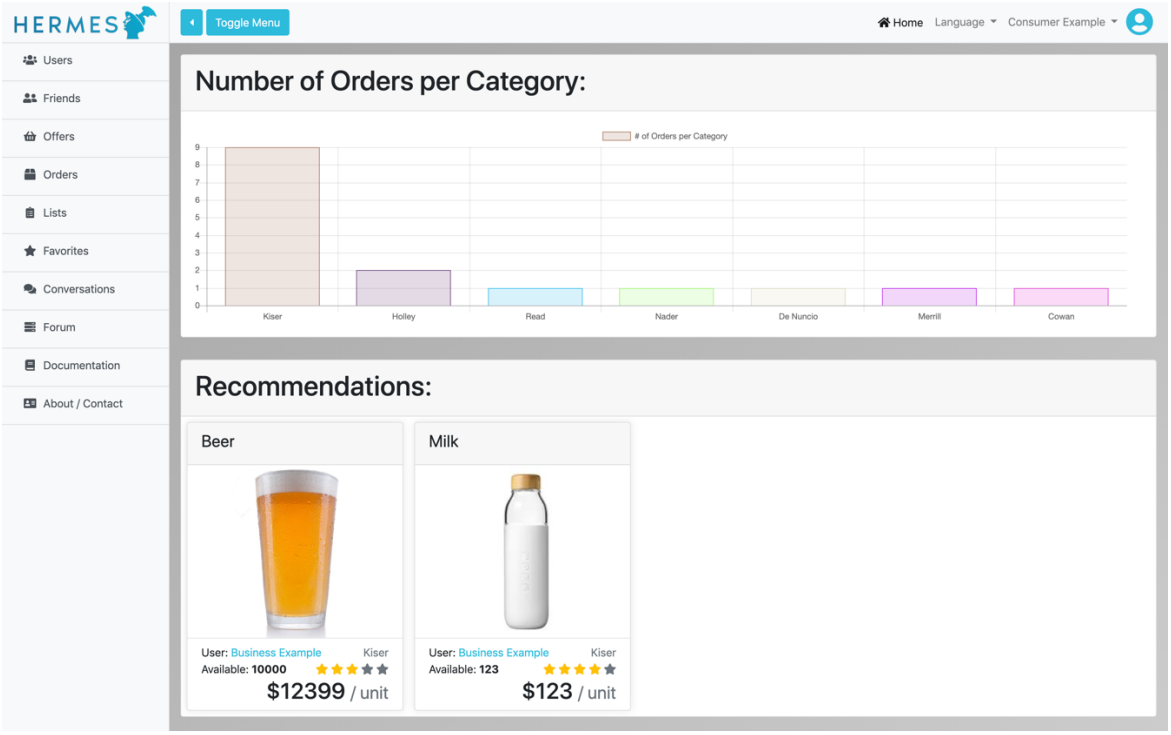
***Picture 14**. Screenshot of User Profile Page.*



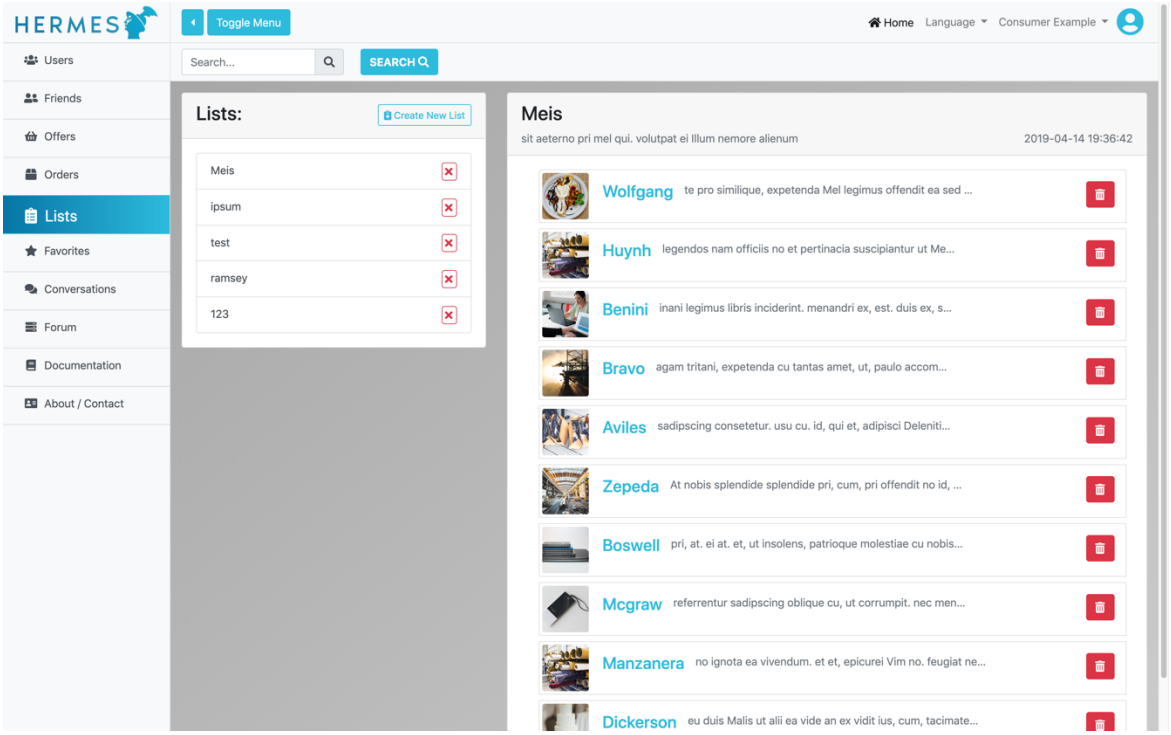
Picture 15. *Screenshot of Offer Page.*



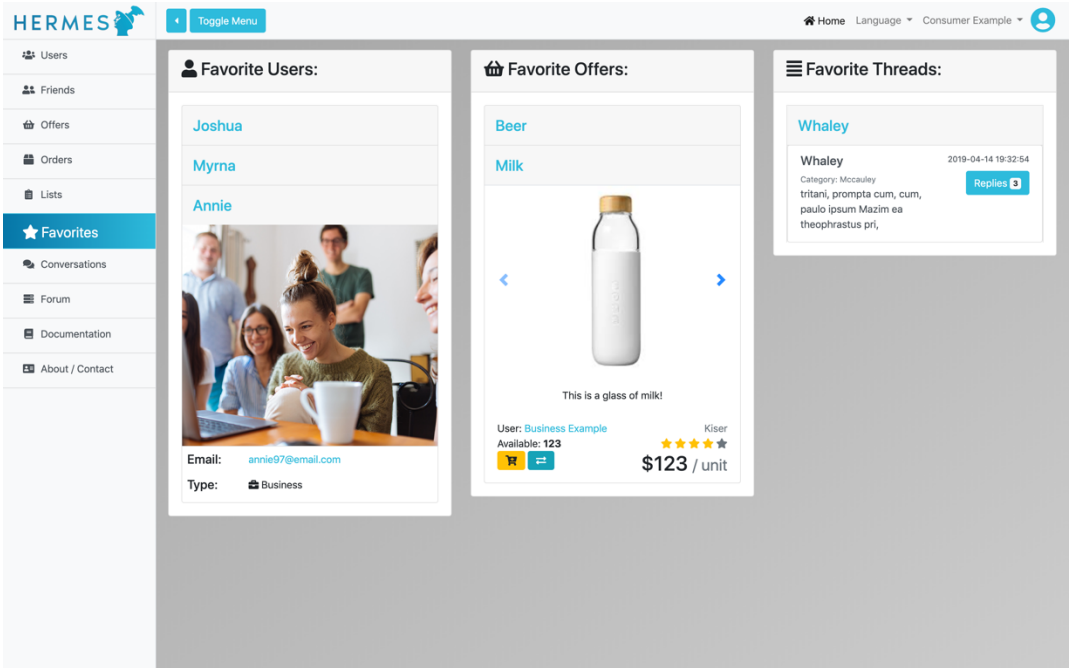
Picture 16. *Screenshot of Orders Page.*



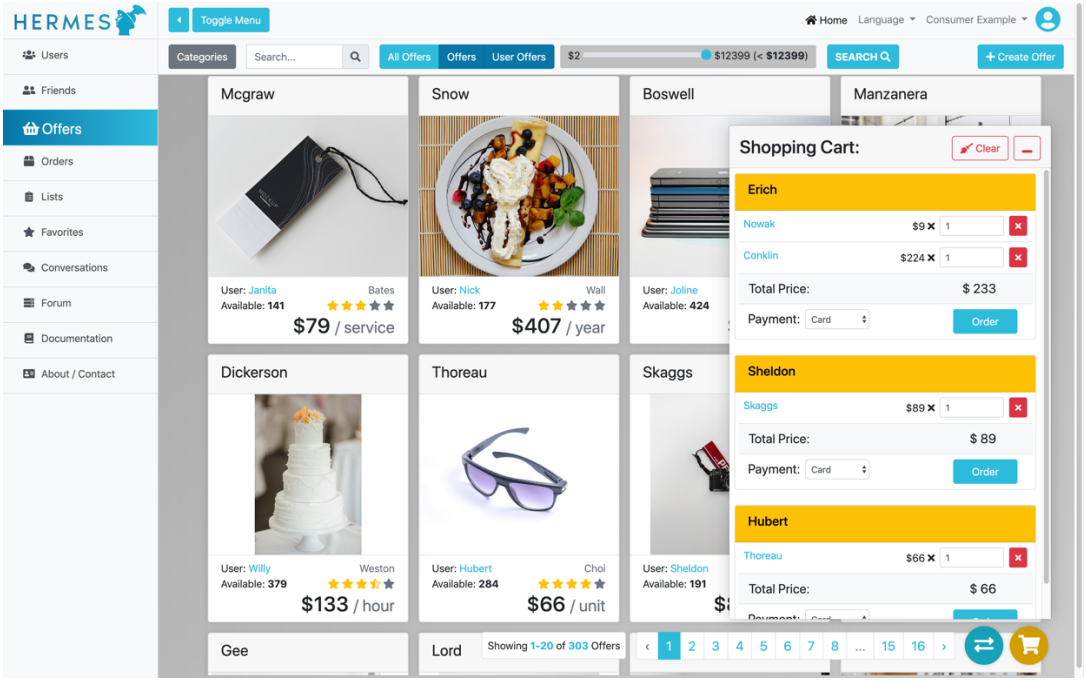
***Picture 17**. Screenshot of Login Page.*



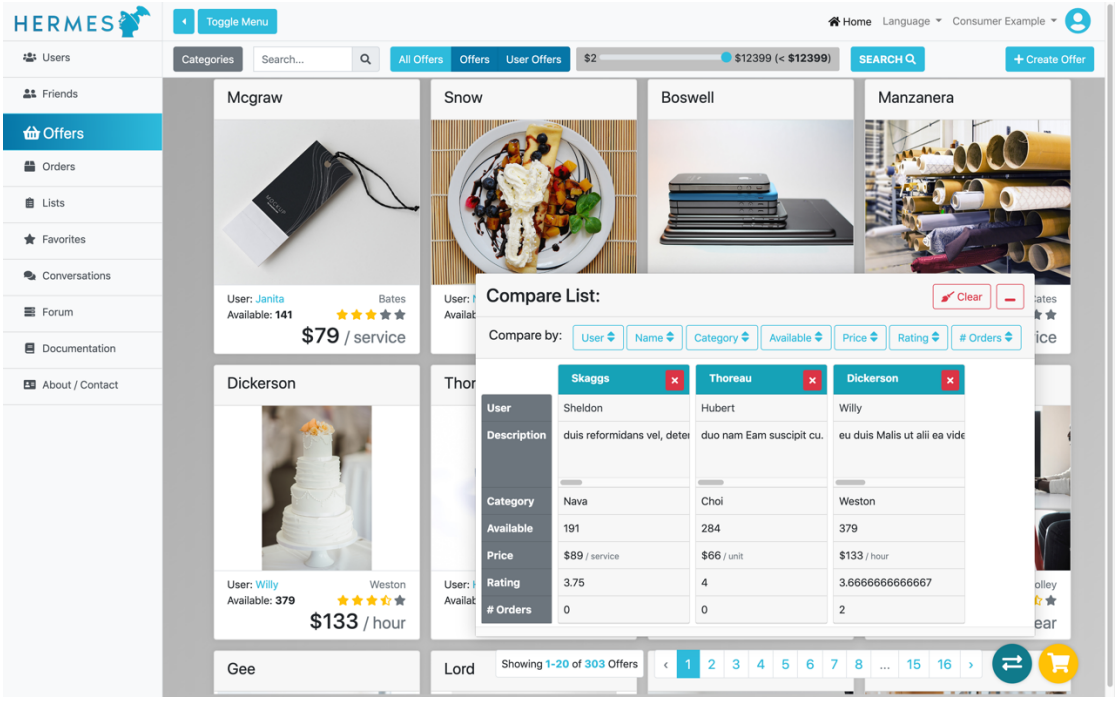
***Picture 18**. Screenshot of Login Page.*



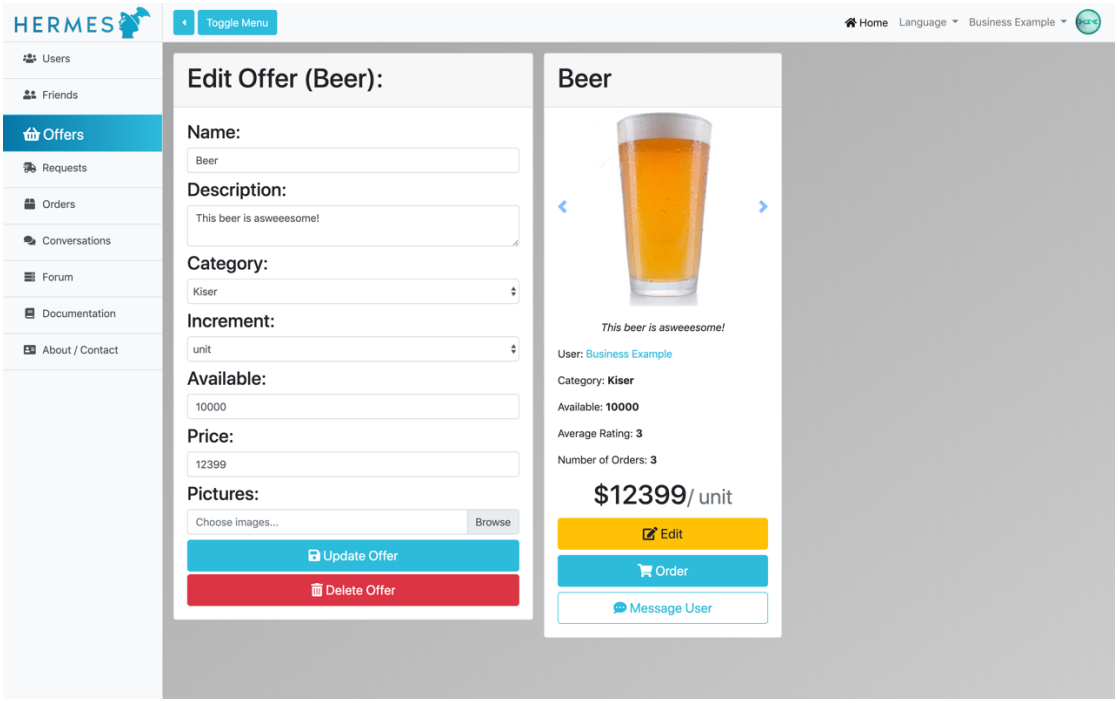
***Picture 19**. Screenshot of Favorites Page.*



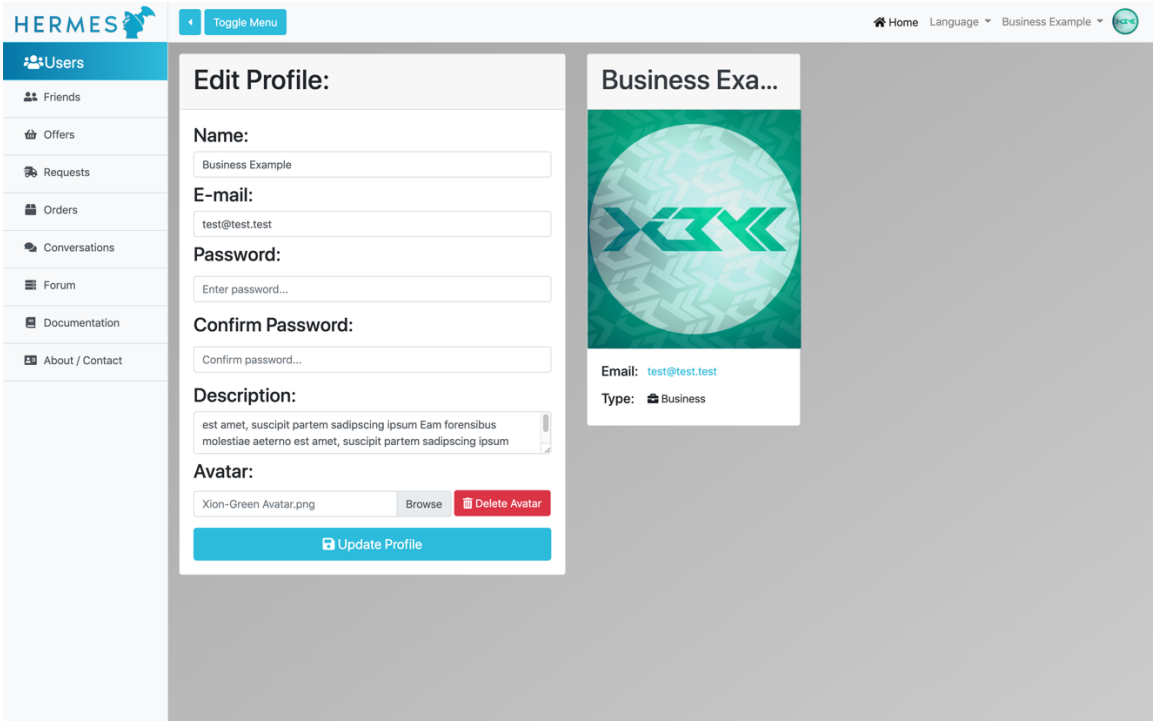
***Picture 20**. Screenshot of Shopping Cart.*



Picture 21. *Screenshot of Compare List.*



Picture 22. *Screenshot of Edit Offer Page.*



***Picture 23**. Screenshot of Edit Profile Page.*

Glossary

Table of figures

Figure	Description
1	*The database design and all of its tables.*

Table of tables

Table	Description
1	*Example of the GANTT chart table with start, finish and duration of each task.*
2	*Results from the RAIL Model Testing*
3	*Results from PHPMetrics regarding the average values*

Table of diagrams

Diagram	Description
1	*Example of MVC flowchart.*

Table of pictures

Picture	Description
1	*The logo of the application.*
2	*Screenshot of Business Home Page.*
3	*Screenshot of Users Page.*
4	*Screenshot of Friends Page with friends' profile open.*
5	*Screenshot of Offers Page.*
6	*Screenshot of Requests Page.*
7	*Screenshot of Conversations Page.*
8	*Screenshot of Forum Page.*
9	*Logo of Laravel Framework.*
10	*Logo of Bootstrap framework.*

11	*Logo of jQuery library.*
12	*The four aspects of RAIL Model.*
13	*Screenshot of Login Page.*
14	*Screenshot of User Profile Page.*
15	*Screenshot of Offer Page.*
16	*Screenshot of Orders Page.*
17	*Screenshot of Login Page.*
18	*Screenshot of Login Page.*
19	*Screenshot of Favorites Page.*
20	*Screenshot of Shopping Cart.*
21	*Screenshot of Compare List.*
22	*Screenshot of Edit Offer Page.*
23	*Screenshot of Edit Profile Page.*

Bibliography

[1] H. Han and S. Trimi, “Social Commerce Design: A Framework and Application,” *Journal of theoretical and applied electronic commerce research*, vol. 12, no. 3, pp. 50–68, Sep. 2017.

[2] Z. Huang and M. Benyoucef, “From e-commerce to social commerce: A close look at design features,” *Electronic Commerce Research and Applications*, vol. 12, no. 4, pp. 246–259, Jul. 2013.

[3] T. Palmere, “How Social Commerce is Changing eCommerce,” *Search Engine People Blog*, 05-Jun-2018. .

[4] N. S. Safa and R. von Solms, “Customers repurchase intention formation in e-commerce,” *South African Journal of Information Management*, no. 1, 2016.

[5] B. Aghamirian, B. Dorri, and B. Aghamirian, “Customer Knowledge Management Application in Gaining Organization’s Competitive Advantage in Electronic Commerce,” *Journal of theoretical and applied electronic commerce research*, no. 1, p. 63, 2015.

[6] J. Wu, M. Xu, Z. Mo, and L. Liao, “The Research of Design Based on Social Commerce,” *International Journal of Social Science Studies*, vol. 3, no. 4, Jun. 2015.

[7] H. Bidgoli, *Electronic Commerce : Principles and Practice*. San Diego: Academic Press, 2002.

[8] “Simple Definition of Social Commerce: Definitive List,” *Social Commerce Today*, 23-Jan-2011.

[9] Claudia Maia, Guilherme Lunardi, Andre Longaray, and Paulo Munhoz, “Factors and characteristics that influence consumers’ participation in social commerce,” *REGE Revista de Gestão, Vol 25, Iss 2, Pp 194-211 (2018)*, no. 2, p. 194, 2018.

[10] Dubovyk Tetiana V. and Ortynska Valentyna V., “The Social Commerce System,” *Bìznes Inform, Vol 3, Pp 325-330 (2017)*, p. 325, 2017.

[11] Y. Baghdadi, “A framework for social commerce design,” *INF SYST*, vol. 60, pp. 95–113, Aug. 2016.

[12] E.-Y. Lee, S.-B. Lee, and Y. J. J. Jeon, “Factors Influencing the Behavioral Intention to Use Food Delivery Apps,” *Social Behavior and Personality: An International Journal*, no. 9, p. 1461, 2017.

[13] S. Molinillo, F. Liébana-Cabanillas, and R. Anaya-Sánchez, “A Social Commerce Intention Model for Traditional E-Commerce Sites,” *Journal of theoretical and applied electronic commerce research*, no. 2, p. 80, 2018.

[14] Maria Cristina ENACHE, “Web Application Frameworks,” *Annals of Dunarea de Jos University. Fascicle I : Economics and Applied Informatics, Vol 21, Iss 3, Pp 82-86 (2015)*, no. 3, p. 82, 2015.

[15] C. S. Miller, J. Zheng, R. Connolly, and A. Olagunju, “Keeping up with web development trends,” in *Proceedings of the 13th annual ACM SIGITE conference on Information technology education - SIGITE ’13*, Orlando, Florida, USA, 2013, p. 59.

[16] J. Feng, X. Fengs, N. Zhang, and J. Peng, “An improved collaborative filtering method based on similarity,” *PLoS ONE*, vol. 13, no. 9, pp. 1–18, Sep. 2018.

[17] Gyöngyvér KOVÁCS, Mihai-Constantin AVORNICULUI, and Botond BENEDEK, “Recommender Systems in E-Commerce Applications,” *Management Intercultural, Vol XVIII, Iss 37, Pp 265-271 (2016)*, no. 37, p. 265, 2016.
[18] H.A.H. Hettiarachchi, C.N. Wickramasinghe, and S. Ranathunga, “The Influence of Social Commerce on Consumer Decisions,” *International Journal of Computational Intelligence Systems , Vol 7, Iss 1, Pp 47-58 (2018)*, no. 1, p. 47, 2018.
[19] M. Quadrana, P. Cremonesi, and D. Jannach, “Sequence-Aware Recommender Systems,” *ACM Computing Surveys*, vol. 51, no. 4, p. 1, Jul. 2018.
[20] P. Morrison, “Building a Security Practices Evaluation Framework,” in *Proceedings of the 2015 Symposium and Bootcamp on the Science of Security*, New York, NY, USA, 2015, pp. 23:1–23:2.
[21] Y.-W. Huang, S.-K. Huang, T.-P. Lin, and C.-H. Tsai, “Web Application Security Assessment by Fault Injection and Behavior Monitoring,” in *Proceedings of the 12th International Conference on World Wide Web*, New York, NY, USA, 2003, pp. 148–159.
[22] Alfonso GONZÁLEZ BRIONES, Pablo CHAMOSO, and Alberto BARRIUSO, “Review of the Main Security Problems with Multi-Agent Systems used in E-commerce Applications,” *Advances in Distributed Computing and Artificial Intelligence Journal, Vol 5, Iss 3, Pp 55-61 (2016)*, no. 3, p. 55, 2016.
[23] R. A. Molla, “Mobile user authentication system (MUAS) for e-commerce applications,” 2017.
[24] Sadiq Ali Khan and Huma Jamshed, “Analysis of Formal Methods for Specification of E-Commerce Applications,” *Mehran University Research Journal of Engineering and Technology, Vol 35, Iss 1, Pp 19-28 (2016)*, no. 1, p. 19, 2016.
[25] Kanar R. Tariq, Ribwar Bakhtyar, David I. Forsyth, and Riyam A. Johni, “Security Aspects in Online Purchasing Applications,” *Kurdistan Journal of Applied Research, Vol 2, Iss 3, Pp 103-105 (2017)*, no. 3, p. 103, 2017.
[26] Claudia Elena Dinucă, “Using Web Mining in E-Commerce Applications,” *Analele Universităţii Constantin Brâncuşi din Târgu Jiu : Seria Economie , Vol 1, Iss 3 , Pp 65 -74 (2011)*, no. 3, 2011.
[27] O. A. I. Gibreel, D. A. AlOtaibi, S. Jeon, and B. Yoo, “Will Insta-Business Be the Electronic Contemporary Bazaar?: An Exploratory Analysis on Electronic Commerce in Kuwait,” in *Proceedings of the 17th International Conference on Electronic Commerce 2015*, New York, NY, USA, 2015, pp. 10:1–10:6.
[28] E. Brown, “Will social commerce ever replace e-commerce?,” *ZDNet*. [Online]. Available: https://www.zdnet.com/article/will-social-commerce-ever-replace-e-commerce/. [Accessed: 01-Dec-2018].
[29] H. Hayder, *Object-Oriented Programming with Php5*. Packt Publishing Ltd, 2007.
[30] J. E. Sweat, *PHP Architect’s Guide to PHP Design Patterns*. Marco Tabini & Associates, Inc., 2005.
[31] K. McArthur, *Pro PHP: Patterns, Frameworks, Testing and More*. Apress, 2008.
[32] “Installation - Laravel - The PHP Framework For Web Artisans.” [Online]. Available: https://laravel.com/docs/5.8. [Accessed: 04-Jul-2019].
[33] “Active Record Basics — Ruby on Rails Guides.” [Online]. Available: https://guides.rubyonrails.org/active_record_basics.html. [Accessed: 04-Jul-2019].

[34] tutorialspoint.com, “Laravel Security,” *www.tutorialspoint.com*. [Online]. Available: https://www.tutorialspoint.com/laravel/laravel_security.htm. [Accessed: 04-Jul-2019].
[35] “SQL Injection.” [Online]. Available: https://www.w3schools.com/sql/sql_injection.asp. [Accessed: 04-Jul-2019].
[36] “Why Laravel is the Recommended Framework for Secure, Mission-Critical Applications,” *Auth0 - Blog*. [Online]. Available: https://auth0.com/blog/why-laravel-is-the-recommended-framework-for-secure-mission-critical-applications/. [Accessed: 04-Jul-2019].
[37] M. O. contributors Jacob Thornton, and Bootstrap, “Introduction.” [Online]. Available: https://getbootstrap.com/docs/4.3/getting-started/introduction/. [Accessed: 04-Jul-2019].
[38] “jQuery Introduction.” [Online]. Available: https://www.w3schools.com/jquery/jquery_intro.asp. [Accessed: 05-Jul-2019].
[39] “Measure Performance with the RAIL Model | Web Fundamentals,” *Google Developers*. [Online]. Available: https://developers.google.com/web/fundamentals/performance/rail. [Accessed: 06-Dec-2018].
[40] tutorialspoint.com, “Load Testing,” *www.tutorialspoint.com*. [Online]. Available: https://www.tutorialspoint.com/software_testing_dictionary/load_testing.htm. [Accessed: 05-Jul-2019].
[41] tutorialspoint.com, “Stress Testing,” *www.tutorialspoint.com*. [Online]. Available: https://www.tutorialspoint.com/software_testing_dictionary/stress_testing.htm. [Accessed: 05-Jul-2019].
[42] Y. Laouris, “The Near Future is Hybrid,” in *Proceedings of the 26th ACM Conference on Hypertext & Social Media*, New York, NY, USA, 2015, pp. 1–1.
[43] J. Leskovec, “New Directions in Recommender Systems,” in *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, New York, NY, USA, 2015, pp. 3–4.
[44] Q. Lu, “Future of the Web and Search,” in *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, New York, NY, USA, 2011, pp. 1–2.