**Tribhuvan University**

**SIDDHANATH SCIENCE CAMPUS**

Institute of Science and Technology

Mahendranagar, Kanchanpur

****

**BURGER ORDERING SYSTEM**

**A PROJECT REPORT**

**Submitted To:**

**Department of Computer Science and Information Technology**

**Siddhanath Science Campus**

**Mahendranagar, Kanchanpur**

***In the Partial Fulfillment of the Requirements for the Bachelor’s Degree in***

***Computer Science and Information Technology***

**Submitted by:**

Bhoj Bahadur Ayer (11269/073)

Bhupendra Rawat (11270/073)

Bhuwan Buda (11271/073)

Prajita Karki (11295/073)

**August, 2021**

**Tribhuvan University**

**SIDDHANATH SCIENCE CAMPUS**

Institute of Science and Technology

Mahendranagar, Kanchanpur



**SUPERVISOR’S RECOMMENDATION**

This is to certify that this project prepared by BHOJ BAHADUR AYER, BHUPENDRA RAWAT, BHUWAN BUDA and PRAJITA KARKI **“BURGER ORDERING SYSTEM**” which is being submitted by to Department of Computer Science & Information Technology, Siddhanath Science Campus in partial fulfillment of the requirements for the award of the degree of Bachelor of Science in Computer Science & Information Technology is prepared under my guidance and supervision.

**......................................................**

**Mr. Rajendra Prasad Joshi**

Supervisor

Department of CSIT

Siddhanath Science Campus

Mahendranagar, Kanchanpur

**Tribhuvan University**

**SIDDHANATH SCIENCE CAMPUS**

Institute of Science and Technology

Mahendranagar, Kanchanpur



**LETTER OF APPROVAL**

This is certify that this project prepared by BHOJ BAHADUR AYER, BHUPENDRA RAWAT, BHUWAN BUDA and PRAJITA KARKI entitled “**BURGER ORDERING SYSTEM**” in partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Information Technology has been well studied. In our opinion, it is satisfactory in the scope and quality as a project for the required degree.

…………………………….. ……………………………

**(Supervisor) (HoD)**

**Mr. Rajendra Prasad Joshi Mr. Keshab Datta Bhatt**

Faculty, Department of CSIT Faculty, Department of CSIT

Siddhanath Science Campus Siddhanath Science campus

Mahendranagar, Kanchanpur Mahendranagar, Kanchanpur

……………………………… ……………………………

**(External Examiner) (Internal Examiner)**

**ACKNOWLEDGEMENT**

Preparing a project report is always a tedious effort. To complete such projects it requires many people to have their support and help. It’s our pleasure to acknowledge the contributions of those people who helped us to complete this project.

We would like to thanks our supervisor **Mr. RAJENDRA PRASAD JOSHI** for guiding and suggesting us the idea of this project work. We would also like to thanks our honorable campus chief **Dr. GAJRAM DAMAI** and our teachers **Mr. KESHAB DATTA BHATTA (HoD), Mr. UPENDRA RAJ JOSHI** for their help and support for shaping up this project.

We, the members of this project would like to thank all the people who have helped us directly or indirectly by their valuable suggestion, contributions and co-operation throughout this project.

**ABSTRACT**

The aim of the project is to build a responsive online web application for burger restaurant which helps customers to order burger online. Beside of other burger ordering systems, this web application will allow customers to select the ingredients they want to have in their burger. Along with this, the application needs to be responsive as the application can be accessed through devices with different size of screens. PHP, MySQL and Bootstrap has been used to develop the backend of the system and React framework with CSS has been used for developing the responsive frontend. After developing the system, different testing methods have been used to find bugs in the system and ensure the quality.

**LIST OF ABBREVIATION**

API : Application Programming Interface

CGI : Common Gateway Interface

CLI : Command-Line Interface

CSS : Cascading Style Sheet

DBMS : Database Management System

DFD : Dataflow Diagram

ERD : Entity Relationship Diagram

GB : Giga Byte

HTML : Hypertext Markup Language

IP : Internet Protocol

MB : Mega Byte

MD5 : Message Digest 5

MVC : Model View Controller

NIC : Network Interface Card

PHP : Hypertext Query Language

RAM : Random Access Memory

SDLC : System Development Life Cycle

SQL : Structure Query Language

TCP : Transmission Control Protocol

UI : User Interface

URL : Universal Resource Locator

**LIST OF FIGURES**

**Name Page Number**

Fig 2.1: Waterfall Model…………………………………………………………...3

Fig 3.1: Use Case Diagram of Burger Ordering System…………………………...8

Fig 3.2: Gantt chart of Burger Ordering System………………………………….11

Fig 4.1: Context diagram of Burger Ordering System……………………………15

Fig 4.2: Level 1 DFD……………………………………………………………..16

Fig 4.3: ER-Diagram……………………………………………………………...17

**LIST OF TABLES**

**Name Page Number**

Table 1: All Database Tables………………………………………………………..18

Table 2: Veg Burger Details………………………………………………………....18

Table 3: Non-Veg Burger Details……………………………………………………19

Table 4: Registered Admins…………………………………………………………19

Table 5: Email Table………………………………………………………………...19

Table 6: User Table…………………………………………………………….........20

**TABLE OF CONTENTS**

SUPERVISIOR’S RECOMMENDATION…………………………………………...i

LETTER OF APPROVAL……………………………………………………………ii

ACKNOWLEDGEMENT…………………………………………………………....iii

ABSTRACT…………………………………………………………………………..iv

LIST OF ABBREVIATION…………………………………………………………..v

LIST OF FIGURES…………………………………………………………………..vi

LIST OF TABLES…………………………………………………………………...vii

TABLE OF CONTENTS…………………………………………………………viii-x

**Chapter 1: Introduction……………………………………….……..1-2**

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

1.2 Problem Definition……………………………………………………............2

1.3 Objectives …………………………………………………………………….2

**Chapter 2: Methodology……………………………………………...3-5**

2.1 Waterfall Model................................................................................................3

2.1.1 Problem Identification………………..……..………………..…………4

2.1.2 Analysis………………………………….……………………...............4

2.1.3 Planning………………………….………….………………………...4-5

2.1.4 Design.......................................................................................................5

2.1.5 Coding……………………………..……….………...............................5

2.1.6 Testing ………………………………………….…................................5

2.1.7 Deployment……………………………………………………..............5

2.1.8 Maintenance………………………………………………….................5

**Chapter 3: Requirement Analysis…………………………………..6-12**

3.1 Requirement Analysis…………………………………………………………6

3.1.1 Functional Requirement…………………………………….................6-7

3.1.2 Non-functional Requirement………………………………………….7-9

3.2 Feasibility Analysis………………………………………………...………9-12

3.2.1 Technical Feasibility………………………………………................9-10

3.2.2 Economic Feasibility…………………………………….......................10

3.2.3 Schedule Feasibility……………………………….………..............10-11

3.2.4 Operational Feasibility………………………………..………………..12

**Chapter 4: System Design…………………………………………………13-20**

4.1 Introduction…………………………………………………………………..13

4.2 Architecture Design…………………………………………………………..13  
4.3 Conceptual Design…………………………………………………………...13

4.4 Logical Design…………………………………………………….................14

4.5 Physical Design………………………………………………………………14

4.6 Structuring System Requirement…………………………………………14-15

4.6.1 Dataflow Diagram…………………………………………………..15-16

4.6.2 ER- diagram…………………………………………………...……….17

4.7 Database Tables………………………………………………………......18-20

**Chapter 5: System Development…………………………………………21-23**

5.1 Development Tools…………………………………………………………..21

5.1.1 Sublime…………………………………………………………………21

5.1.2 Visual Studio Code……………………………………………………..21

5.1.3 My SQL………………………………………………….......................21

5.1.4 XAMPP Server……………………………………………………..21-22

5.2 Programming Tools……………………………………………………….22-23

5.2.1 Front-End Tools………………………………………………………..22

5.2.2 Back-End Tools………………………………………………………...23

**Chapter 6: Implementation and Testing…………………………..24-26**

6.1 Implementation………………………......…………………………………..24

6.2 Testing………………………………………………………..………………24  
 6.2.1 Unit Testing……………………………………………………………25

6.2.2 Integration Testing……………………………………………………..25

6.2.3 System Testing…………………………………………………………25

6.2.4 Module Testing………………………………………………………...25

6.2.5 Acceptance Testing…………………………………………………….25

6.3 Result Analysis……………………………………………………………….26

6.3.1 Validation…………………………………………………...................26

**Chapter 7: Conclusion and Recommendation……………………….27**

7.1 Achievement of the Project………………………………………………….27

7.2 Limitation of the System…………………………………………………….27

7.3 Future Recommendation…………………………………………………….27

7.4 Conclusion…………………………………………………………………...27

**APPENDIX I (Snap shots)...................................................................................28-35**

**REFERENCES...........................................................................................................36**

**CHAPTER 1**

**INTRODUCTION**

* 1. **. Introduction**

This system is very basic need for those restaurants or corners which are meant to only sell the burger. We may find many more of the burger ordering sites but this one is different from them.

Burger Ordering System does not contains static or readymade burgers instead it allows customers to make their own. In most cases, customers don’t get the exact burger of their need, for example the amount of ingredients added in the burger, the price of the burger, etc. To overcome such problems this system is built. It provides new technique for making order.

The objective of **Burger Ordering System** is to allow people to interact with the system with its easy-to-use GUI system. Due to its simple and responsive GUI, most of the people would easily interact with the system. There are no such zig-zags in this system.

* 1. **. Problem Definition**

By doing researches and study, we came to know that the places where burger are sold, maybe restaurants, corners, café, etc. , have some issues in getting orders. People should have to visit those places for ordering the burger and getting their burger. There are some problems of traditional burger ordering system, which results in time consumption and manpower problems. Those problems are:

* People have to visit restaurants to place their order and make payment.
* Due to the traditional system, restaurants needs manpower for managing the order and payment. Sometimes those restaurants don’t find employees in time.
* To solve these problems, a burger order system has been developed which is originally designed for small scale business. But this system is only applicable in restaurants where we order only burger. It is especially designed for ordering burger. The benefits of the burger ordering system are:
* People can make order from their home instead of going to the restaurants.
* By using this system, time and labor cost are managed.
* This will help reduce human intervention that employees face in restaurants.
* The system will be easily available since it is a web application which resides in internet.

**1.3. Objectives**

The aim in developing **Burger Ordering System** is:-

1. To provide an easy way to place order of burger.
2. To bring the best way of allowing customers to select the ingredients they need to have in their burger.
3. To manage the orders of the customers in very simple and easiest way.
4. To make a responsive system which can be accessed from small mobile devices to large laptop/desktop devices.

**CHAPTER 2**

**METHODOLOGY**

This chapter of the report aims to provide a clear vision on methods and methodologies used for this project. In first, it describes the methodology and methods selected for the research and give rational why these methods and methodology have been used. Then it discusses about software development methodology and give rational for it.

**2.1. Waterfall Model**

There are different models for research. Among them, waterfall model is chosen, as it is easy to understand and implement. Many beginners use this methodology for research. Hence, this methodology is used for research in this project. Following steps have been used to formulate project.

Identify problem

System analysis

Planning

Design

Coding

Testing

Deployment

Maintenance

Fig.2.1: Waterfall Model

* + 1. **Problem Identification**

This is the first step of the research. In this step, different topics and subject area has been reviewed to find problem. After setting the goal to developing restaurant system, the study has been narrowed to online food ordering system.

Once the goal was set, data collection process began to get information about the online food ordering system history and past works on this domain.

Following are some problems found in existing system:

* Customers generally can’t view the ingredients and direction of prepare.
* There is no provision of digitalization while ordering food.
* Restaurants manage their business by manually.
* Quite expensive for customers.
* Unaware of menu items.
  + 1. **Analysis**

Analysis is done to find out that the development of project is worthy or not. If the project goes in loss or it cannot fulfill user’s requirements then it is not worthy to develop the project. Feasibility study is also done in this phase to determine if project is cost effective, if there are sufficient technicals or if it legal or operational and many more analysis are done.

* + 1. **Planning**

In planning phase, to develop a new system which is a first step is to identify a need for the “Burger Ordering System” and also plan how to develop the functional requirements of a system. This will include determining whether a business problem or opportunity exists, conducting a feasibility study to determine the developing a project plan.

**2.1.4 Design**

Design is the process of deciding the workflow of the project and how to work on it. For designing process different tools are used here we are using context diagram, data flow diagram and entity relationship diagram.

* + 1. **Coding**

Coding is done according to design here we used React and CSS for frontend coding and PHP and SQL for Backend coding. Visual Studio Code editor is used to code the project.

* + 1. **Testing**

Testing is done to see if the project is working as user requirement or not, or is it working as coded or not. Some testing performed in this project are Black Box testing, White Box testing, Unit testing, Integration testing and acceptance testing.

**2.1.7 Deployment**

After the project is ready it is deployed to the market.

**2.1.8 Maintenance**

Maintenance is the process that is done after deployment of the project.

If some problem occurs while working of the system then it is needed. Since our project is not deployed it is not necessary right now.

**CHAPTER 3**

**SYSTEM REQUIREMENT**

**3.1. Requirement Analysis**

Requirement analysis, also called requirements engineering, is the process of determining user expectations for a new or modified product. These features are called requirements, must be quantifiable, relevant and details. In software engineering, such requirement are often called functional specifications.

There are two different types of requirement analysis, they are as follows:

**3.1.1 Functional Requirement**

Functional requirements include the functions performed by specific screens, outlines of work-flows performed by the system and other requirements that the system must meet.

Developed system should have following functional requirements:

* System should have the functionality of recommendation system.
* System should provide required information to the customers.
* System should be easily access by the users.

Functional requirement has been illustrated by following use case diagram:

Customer

Fig (a): Customer Use Case Diagram

Admin

Fig (b): Admin Use Case Diagram

**Fig 3.1: Use Case Diagram for Burger Ordering System**

**3.1.2 Non-functional Requirement**

In addition to the obvious features and functions that we will provide in our system, there are other requirements that don’t actually do anything, but are important characteristics nevertheless which are called non-functional requirements or sometimes Quality Attributes. For example, attributes such as performance, security, usability, compatibility. We can’t write a specific line of code to implement them. The specification needs to describe any such attributes the user requires. We must decide the kind of requirements that apply to our project and include those that are appropriate. Each requirement is simply stated in human readable format and each requirement must be objective and quantifiable. There must be some measurable way to access whether the requirement has been met.

Some examples of non-functional requirements are given below:

* **Performance Requirements**

Requirements about resources required, response time, transaction rates, or anything else that is done with performance is the performance requirement.

* **Operating Constraints**

Operating constraints lists any non-time constraints. This could include system resources, people, needed software etc.

* **Platform Constraints**

Platform Constraints discuss about the target platform. We have to be as specific or general as the user requires. If the user doesn’t care, there are still platform constraints.

* **Accuracy and Precision**

Accuracy and Precision discusses about the requirements about the accuracy and precision of the data.

* **Modifiability**

Modifiability discusses about the requirements about the effort required to make changes in the software.

* **Portability**

Portability tells about the effort required to move the software to a different target platform.

* **Reliability**

Reliability tells about the requirements about how often the software fails. The measurement is often expressed in mean time between failures. We have to specify the consequences of software failure, how to protect from failure, a strategy for error detection and a strategy for correction.

* **Security**

One or more requirements about protection of our system and its data are making our data more secure. The measurement can be expressed in a variety of ways like effort, skill level, time etc.

* **Usability**

Usability discusses about the requirements about how difficult it will be to learn and operate the system. The requirements are often expressed in learning time or similar matrices.

* **Legal**

There may be legal issues involving privacy of information, intellectual property rights, export of restricted technologies etc.

The non-functional requirements include following:

* The recommendations should be fast.
* The system should provide right information.
* The user interface of the system should be easy to use i.e. the system should be more user friendly.

**3.2. Feasibility Analysis**

Feasibility study is the study of system about its ability to meet the user requirement. Feasibility is the study of impact, which happens in the organization by the development of a system. It is wise to think about the feasibility of any problem this project looks to solve or undertake. The impact can be either positive or negative. When the positives nominate the negatives, then the system is considered feasible. The main purpose of the feasibility study is check that the system is feasible or not. Our system must be feasible so that it can be implemented in time and used easily in various browsers. The main feature required for project to be a feasible are of mainly four types which are as follows:

**3.2.1 Technical Feasibility**

Technical feasibility study is the complete study of the project in terms of input, processes, output, fields, programs, and procedures. It is a very effective tool for long term planning and trouble shooting. The technical feasibility study should most essentially support the financial information of an organization.

Developed system is technically feasible. To use the system, using system doesn’t require advanced technical knowledge. It’s simple UI makes interaction easy. Anyone who has

basic knowledge about computers can use this system. The entire system can be developed with the presently resource.

**Hardware Requirement (For Implement This System)**

RAM : 2 GB

Free Hard Disk Space : 20 GB

Operating System : Window XP or higher

**Software Requirement (For Implement This System)**

Operating System : Any Windows Operating System.

Language : PHP and React for implementing the System.

Server : XAMPP Server and Node JS development server.

Browser : Chrome/Firefox/Microsoft Edge.

**3.2.2 Economic Feasibility**

Economic feasibility analysis is the most commonly used method for determining the efficiency of a project. It is also known as cost analysis. It helps in identifying profit against investment expected from a project. Cost and time are the most essential factors involved in this field of study. Developed system is economically feasible. It can be developed on simple PC which can be available in affordable cost.

**3.2.3 Schedule Feasibility**

Schedule Feasibility is defined as the probability of a project to be completed within its scheduled time limits by a planned due date. If a project has a high probability to be completed on time, then its schedule feasibility is high. In many cases, a project will be unsuccessful if it takes longer than it was estimated. Some external environmental conditions may change. Hence, a project can be losing its benefits, expediency and profitability. If a work to be accomplished at a project does not fit the timeframes demanded by its users or customers, then a schedule is unfeasible i.e. amount of work should be reduced or other schedule compression methods applied.

If the project managers want to see their projects completed before they can lose their utility, they need to give proper attention to controlling their schedule feasibility. To calculate and continually re-examine whether it is possible to complete all amount and scope of work lying ahead, utilizing the given amount of resources, within required period of time.

Gantt chart for the project timeline is shown below:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Nov 2020 | | |  |  | Dec 2020 | |  | Jan 2021 |
| 15-19 | 20-22 | 23-30 | 1-2 | 3-7 | 8-9 | 10-21 | 22-31 | 1-9 |
| Proposal Design |  |  |  |  |  |  |  |  |  |
| Setting Lab |  |  |  |  |  |  |  |  |  |
| Source code analysis |  |  |  |  |  |  |  |  |  |
| Information gathering |  |  |  |  |  |  |  |  |  |
| Automated Scanning |  |  |  |  |  |  |  |  |  |
| Manual Scanning and Testing |  |  |  |  |  |  |  |  |  |
| Gaining Access |  |  |  |  |  |  |  |  |  |
| Documenting |  |  |  |  |  |  |  |  |  |

**Fig 3.2: Gantt chart of Burger Ordering System**

**3.2.4 Operational Feasibility**

Operational feasibility is the measure of how well a proposed system solves the problems, and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development.

By providing the simple user interface, the system helps any users to use our system. There are no any complex languages, layout and views used so, normally anyone can easily use the system.

**CHAPTER 4**

**SYSTEM DESIGN**

**4.1. Introduction**

System design is the process of defining the architecture, components, modules, interfaces and data for a system to satisfy specified requirements of the system. System development is the process of creating along with the process, models and methodologies used to develop them. The goal of the produced model is call design of the system.

In this phase the focuses on the detail implement of system are recommended, requirement specifications are translated into design specification. In this phase the analyst develop the detailed design of system. System design goes through different phase of development.

**4.2. Architectural Design**

A systems Architect responsibility includes the ability to create, review, and update designs or blueprints to provide an overall direction for the system, project, department, or enterprise. These fundamentals of system design can go through various stages from the initial project concept to the final plan prior to implementation. Not only is it important to develop a design or roadmap at each stage but it can also be a useful tool for training and marketing of the concept. The age old saying “a picture is worth a thousand words” still holds true. Each stage builds upon the previous stage in detail allowing the individual to more completely understand the big picture and then focus on the details as it progresses.

**4.3. Conceptual Design**

A conceptual design is an abstract or high level design which includes only the most important components and entities. The main goal of conceptual design is to provide an understandable picture of the overall purpose of the proposed solution. Components may include major technology systems, external systems that are required for integration or overall functionality, high level data flow, and system functionality.

**4.4. Logical Design**

A logical design is a more detailed design which includes all major components and entities plus their relationships. The data flows and connections are detailed in this stage. The target audience is typically developers or other systems architects. However, it is possible to create logical designs for business purposes to ensure that all components and functionality is accounted and well understood. Logical design does not include physical server names or addresses. They do include any business services, application names and details, and other relevant information for development purposes.

**4.5. Physical Design**

A physical design has all major components and entities identified within specific physical servers and locations of specific software services, objects, or solutions. Include all known details such as operating systems, version numbers, and even patches that are relevant. Any physical constraints or limitations should also be identified within the server components, data flows, or connections. The UML (Unified Modeling Language) is also another method that can be explored for design and definition for these and other designs. There are several diagram types within the UML which could be developed at each one of these design stages.

**4.6. Structuring System Requirement**

In structuring system requirements, for getting the better idea about the process of the system and to know the actual mechanism of the system by giving the structure to the system requirement is done. For showing the flow of the system and to represent the system, we have to design the different models such as Flow-Chart Diagram, ER-Diagram and DFD (Data Flow Diagram).

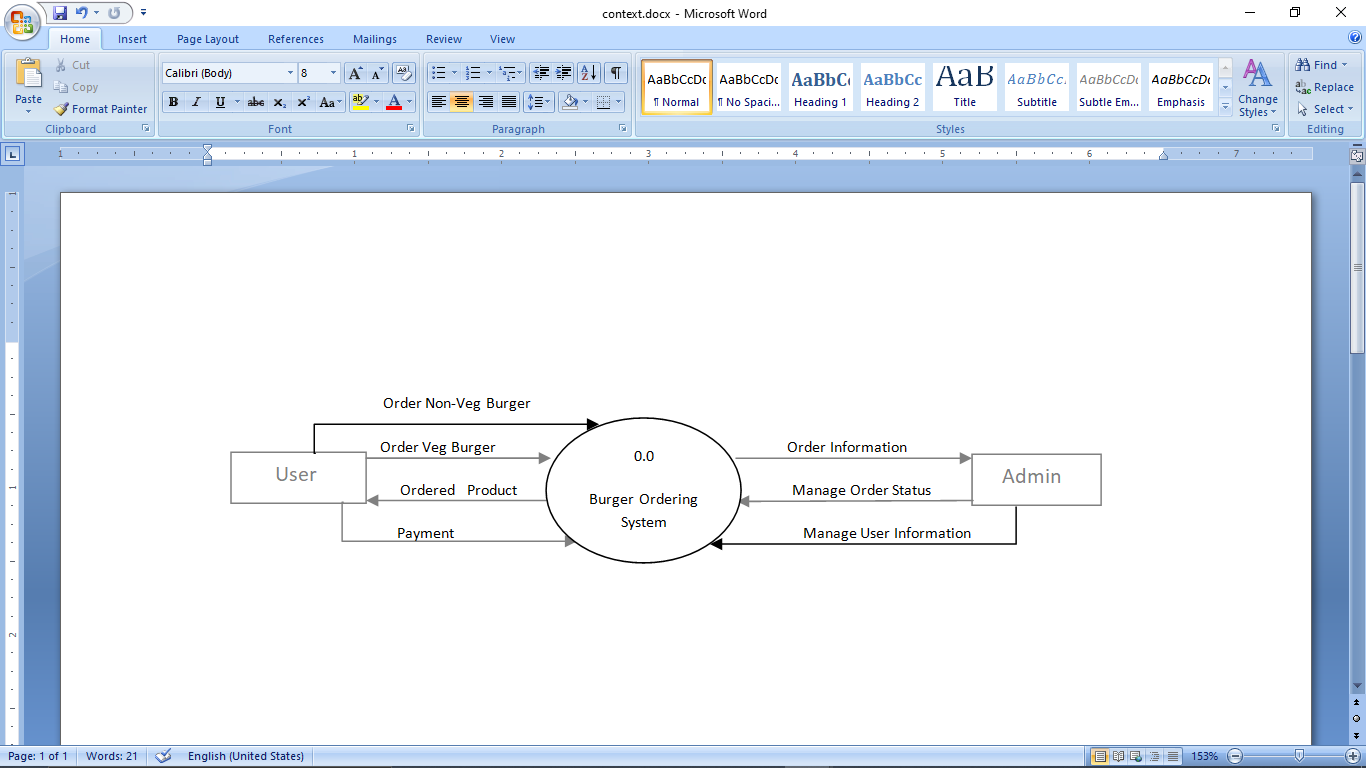
In this project, we performed various activities regarding the relationships, data manipulations and data processing. We created relationships between two or more entity sets so that they can provide better information while searching result in the system. The data processing is represented in the diagrammatic form so that every individual who look or seek at the diagram can understand about the main theme or motto of this system. By following the Structured Approach, the diagrams are made that contains ER-Diagram (Entity Relationship Diagram) as a data modeling approach and DFD (Data Flow Diagram) as a process modeling approach.

**4.6.1 Data flow diagram**

A data flow diagram (DFD) is a graphical representation of the “flow” of data through an information system, modeling the system process. A DFD is often used create an overview of the system. It enables us to represent the processes in our information system from the viewpoint of data. The DFD lets us visualize how the system operates, what the system accomplishes and how it will be implemented, when it is refined with further specification. DFD is one of the most important modeling tool. It is used to model the system, component that interact with the system, uses the data and information flows in the system.

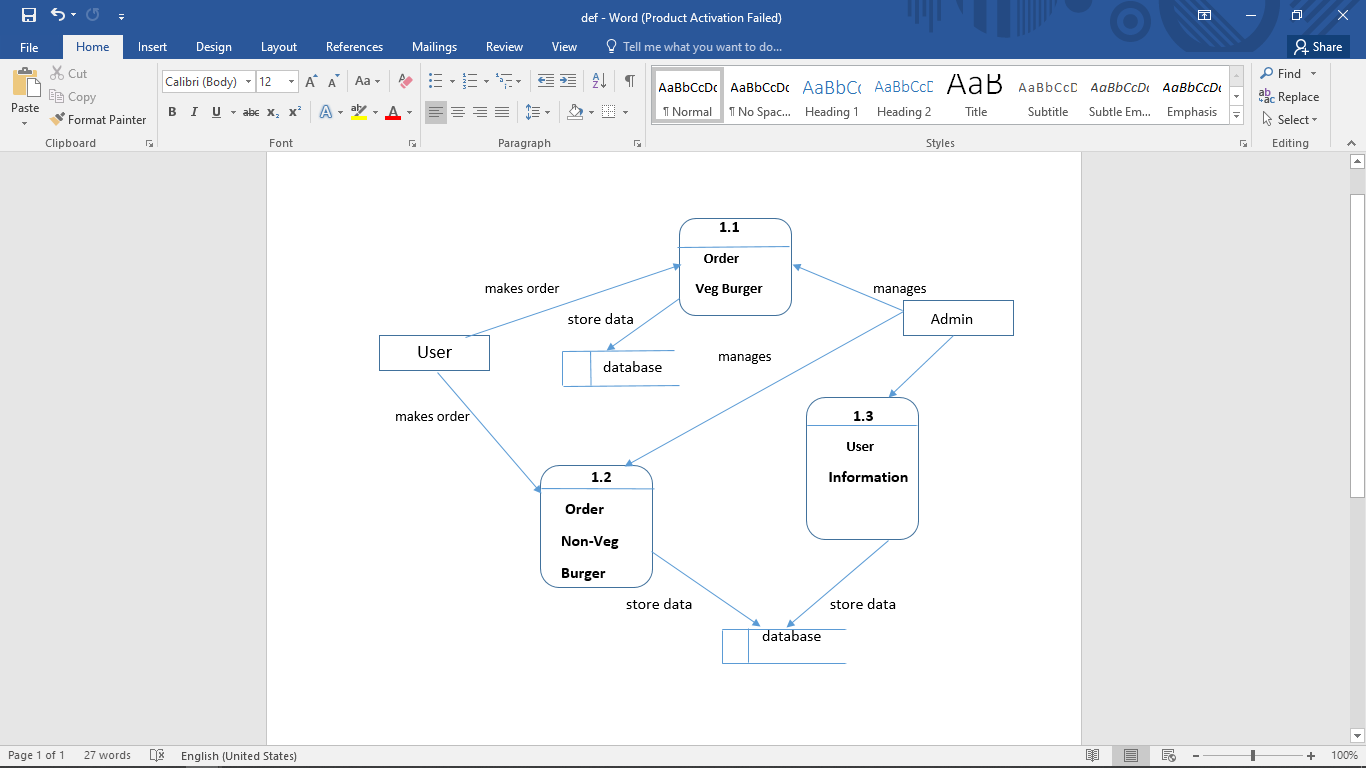
**Context Diagram**

The Context Diagram shows the system under consideration as a single high-level process and then shows the relationship that the system has with other external entities such as systems, organizational groups, external data stores etc. The flow of the system is given below.



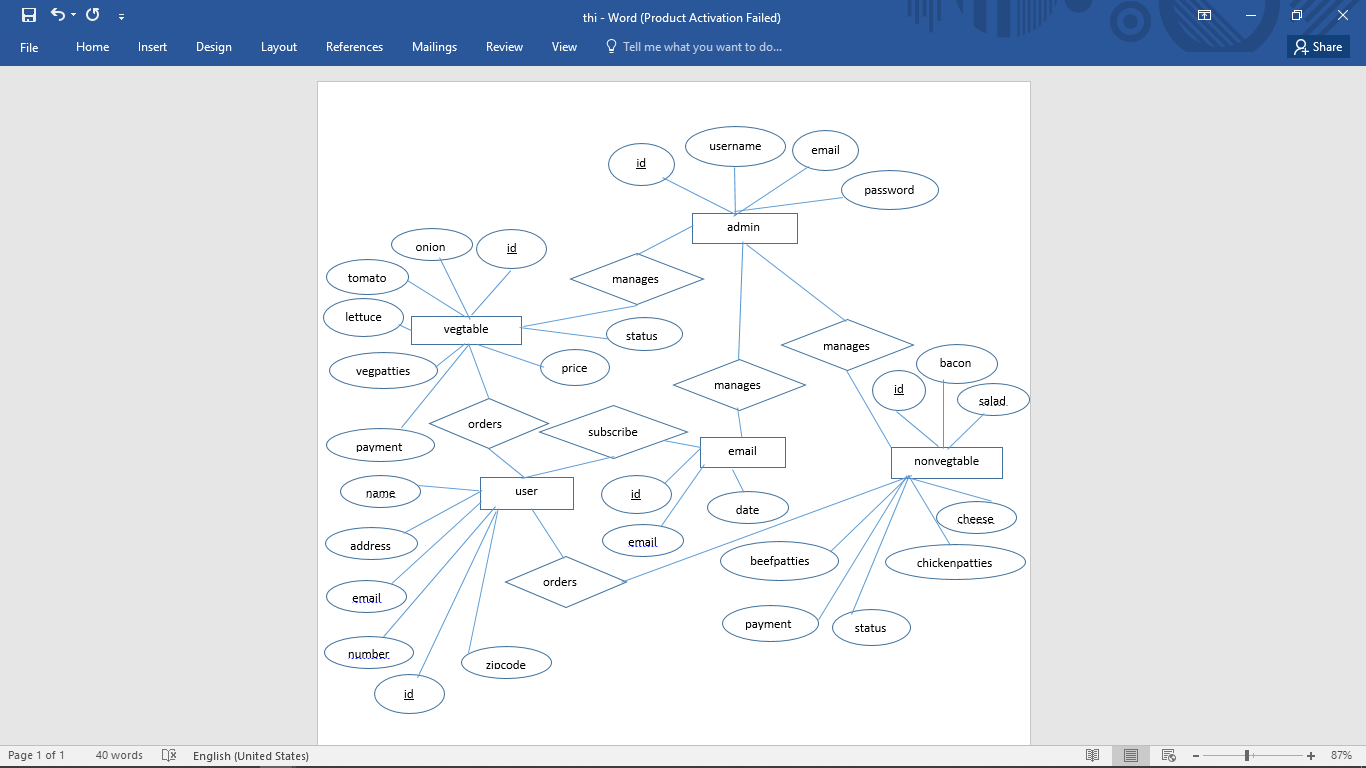
**Fig 4.1: Context diagram of Burger Ordering System**

**Level-1 Data Flow Diagram**

****

**`Fig 4.2: Level-1 Dataflow Diagram of Burger Ordering System**

**4.6.2 ER-Diagram**

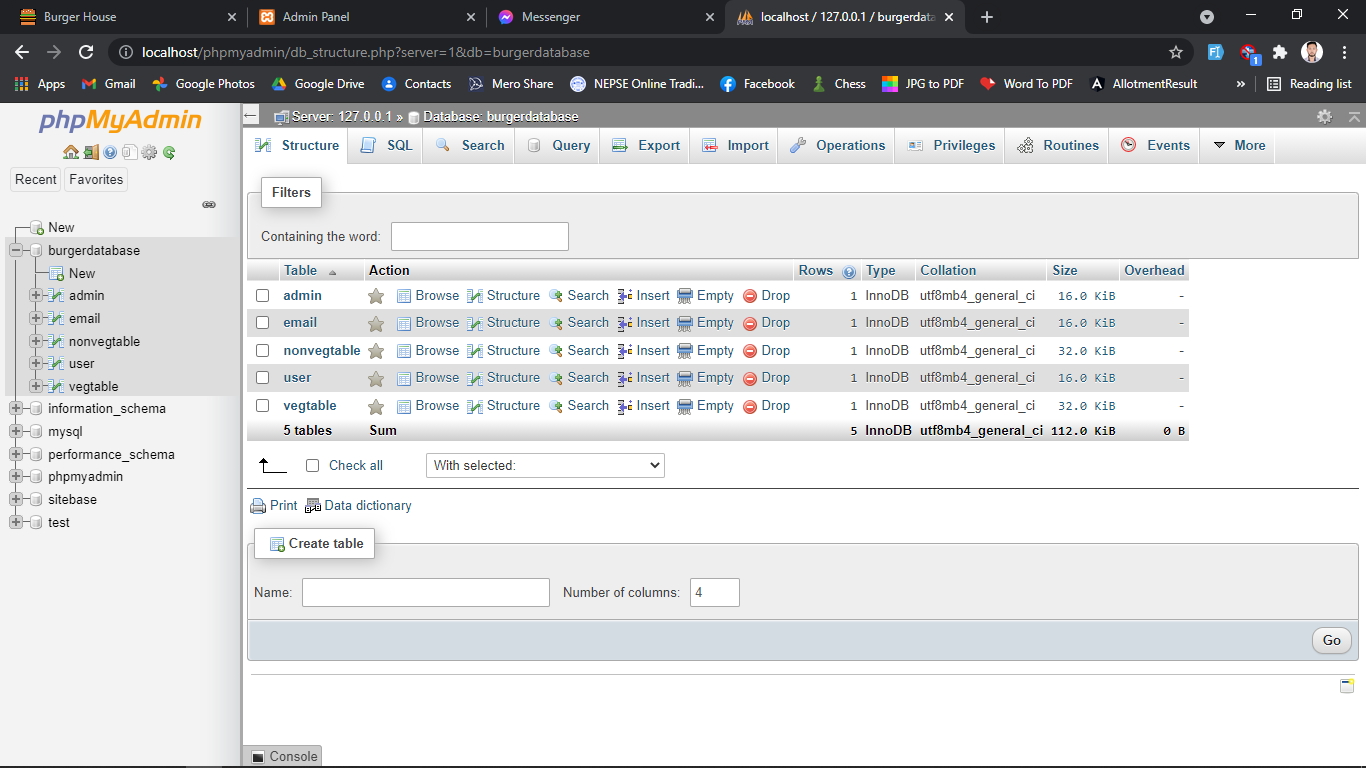
****

­­

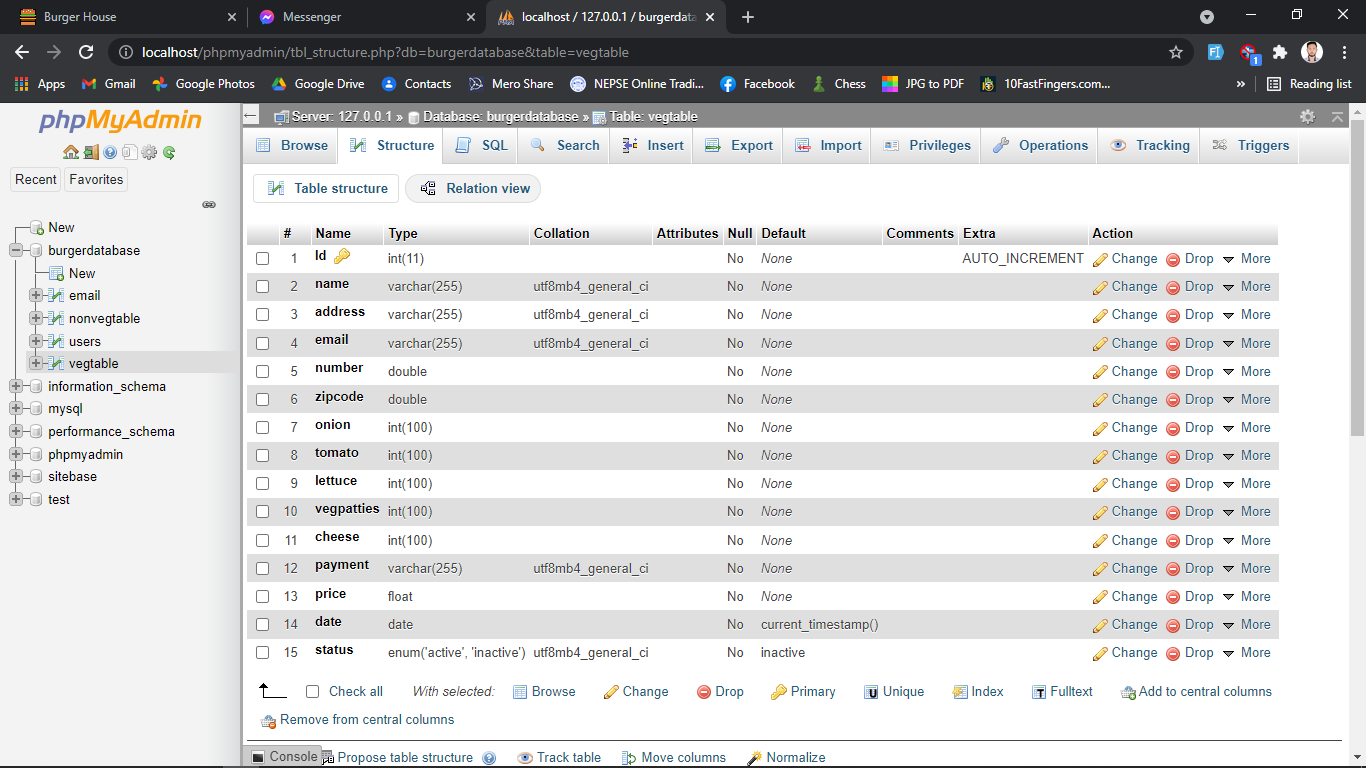
**Fig 4.3: ER-diagram of Burger Management System**

**4.6.3 Data Table**

A table is a collection of related data held in a structured format within a database. It consists of columns, and rows. In relational databases and flat file databases, a table is a set of data elements (values) using a model of vertical columns (identifiable by name) and horizontal rows; the cell is being the unit where a row and column intersect. A table has a specified number of columns, but can have any number of rows. Each row is identified by one or more values appearing in a particular column subset. The columns subset, which uniquely identifies a row, is called the primary key.



*Table 1: All Databases*



*Table 2: Veg Burger Table*

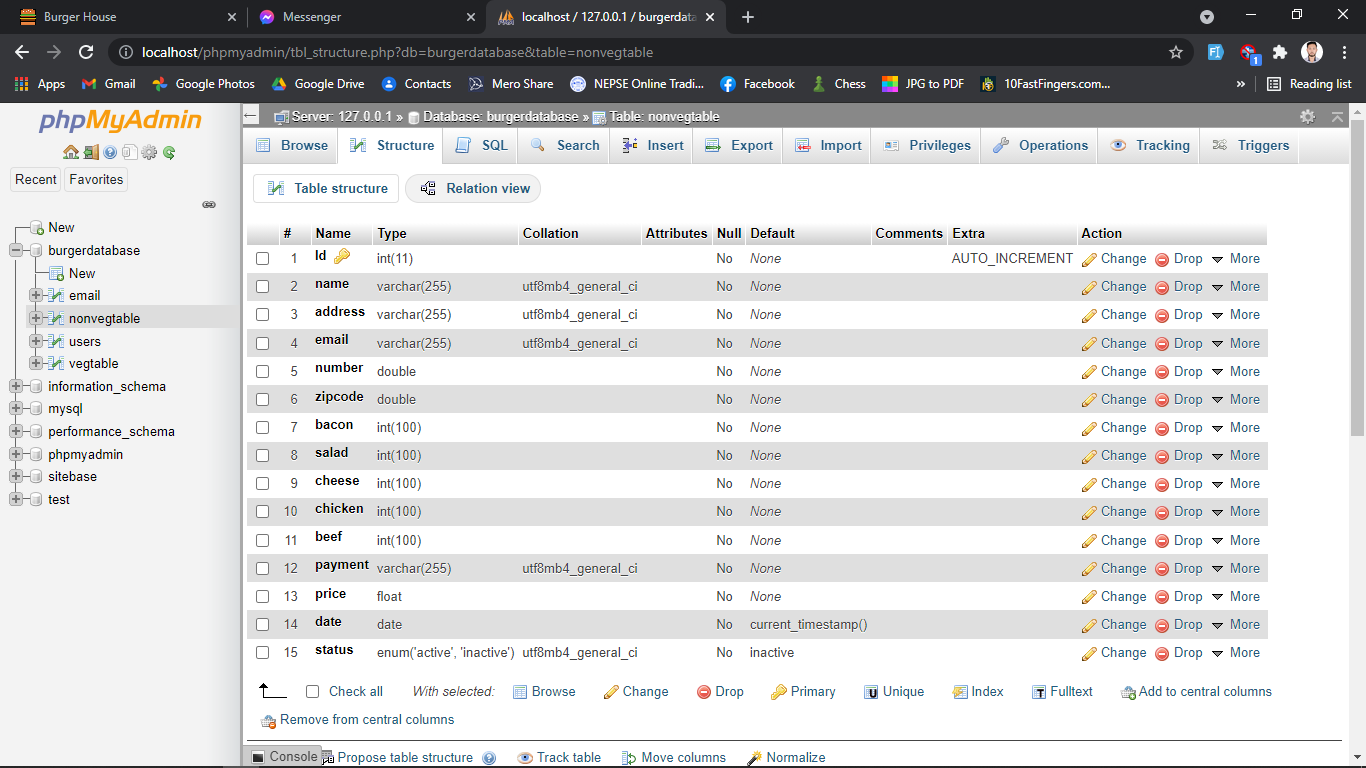


Table 3: Non-veg Burger Table

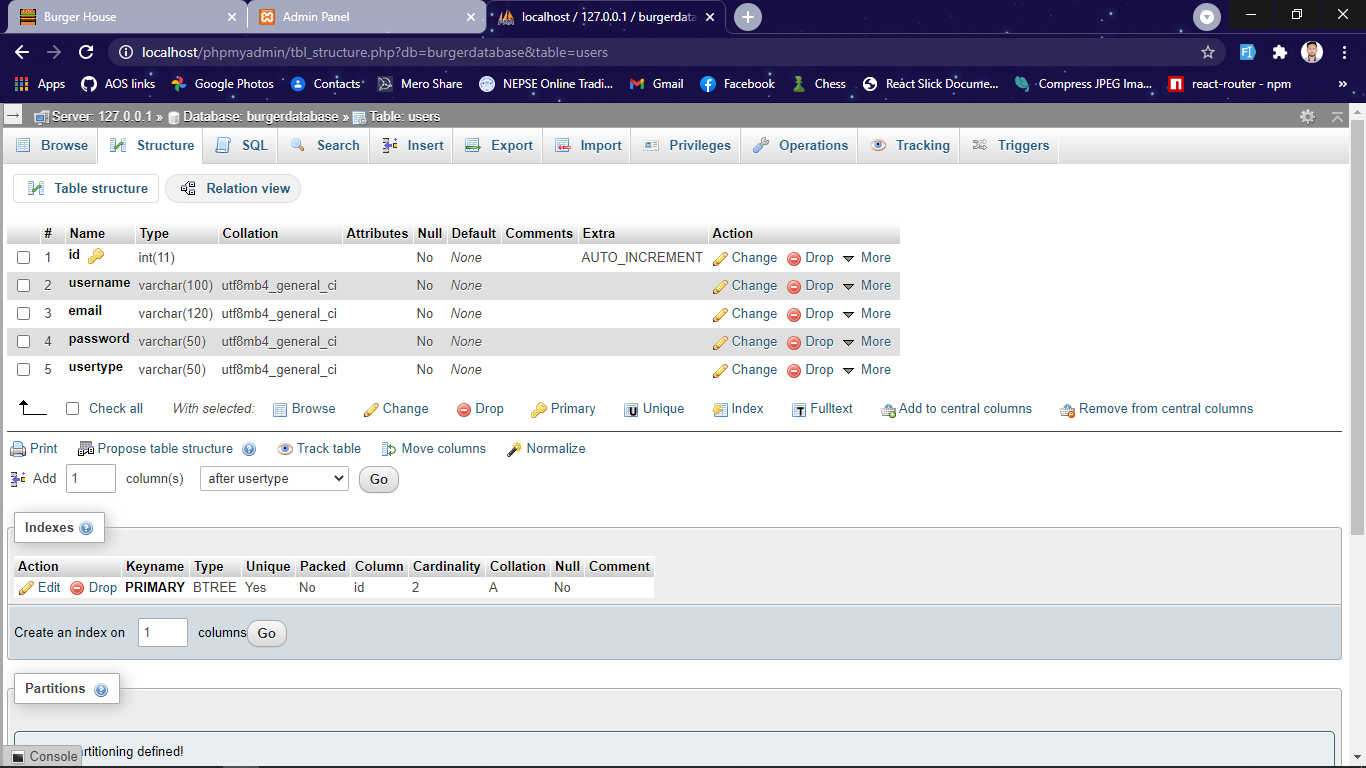
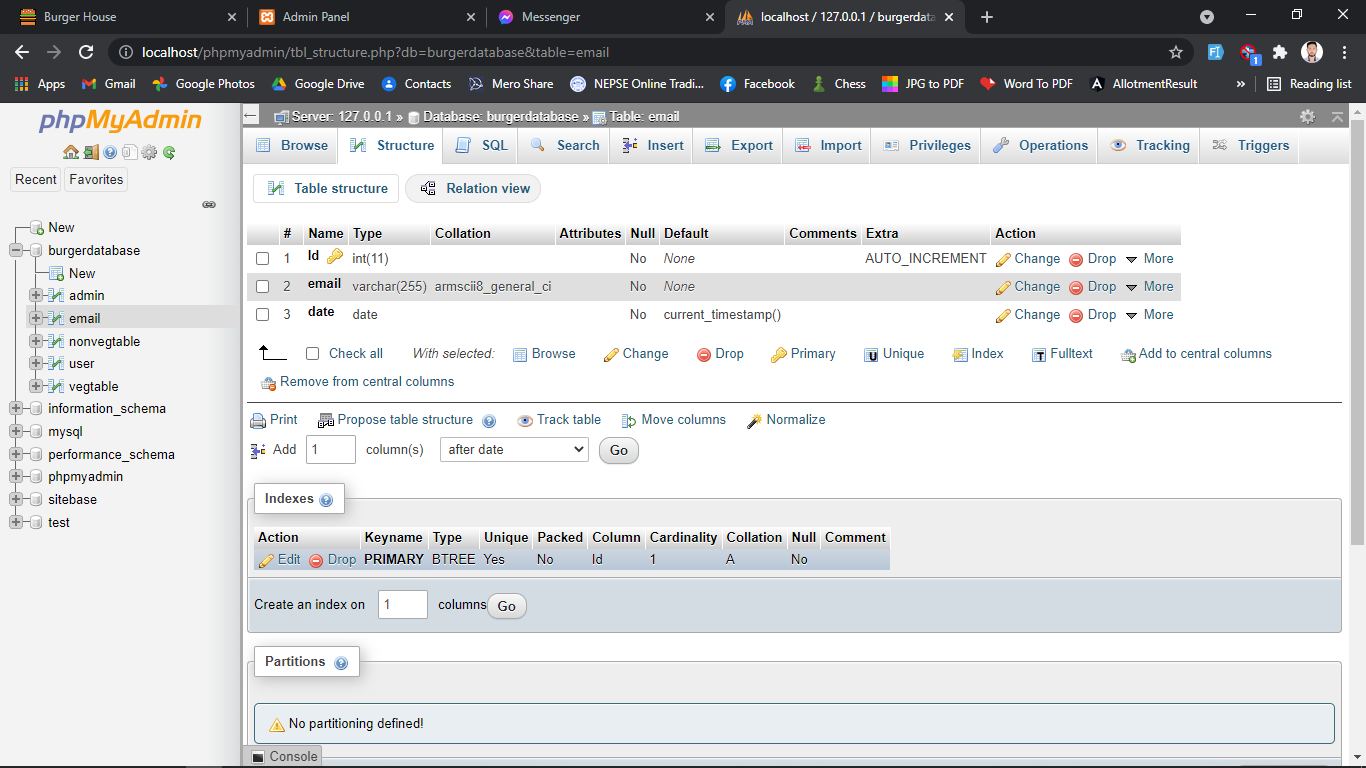
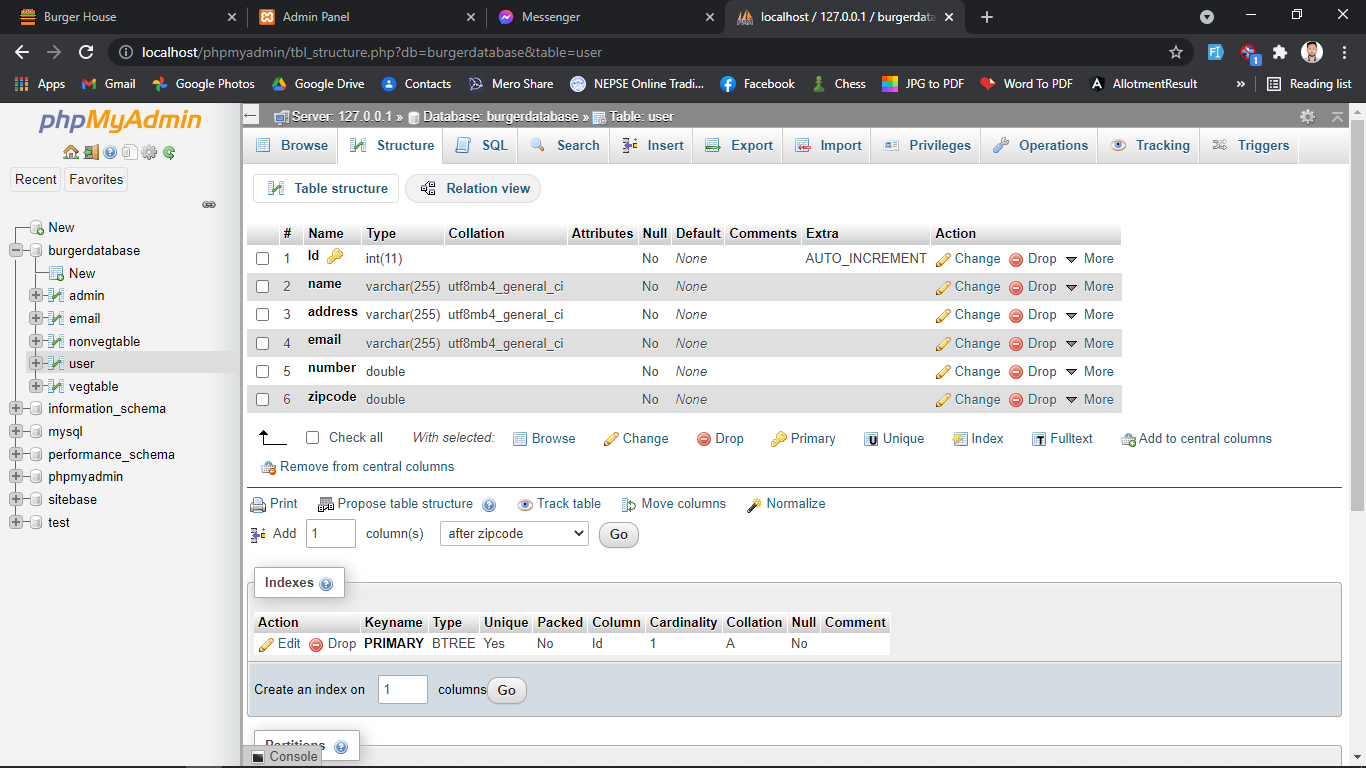


Table 4: Registered Admin



*Table 5: Email Table*

**

*Table 6: User Table*

**CHAPTER 5**

**SYSTEM DEVELOPMENT**

**5.1. Development Tools**

As this is web based project, so the software which are used to develop the web site are web based and the backend for the storing the database of the site. Technology and software for the sake of the development of this project are as follows:

**5.1.1 Sublime Text**

For coding, we have used popular Sublime called Sublime text which makes easier to do tasks, have better interface and easy for communication with the different sub folders and pages.

**5.1.2 Visual Studio Code**

Visual Studio Code is a source code editor developed by Microsoft for Windows, Linux and mac OS. It is a super-fast and lightweight Source Code editor which can be used to view, edit, run and debug source code for applications. It is mainly used by front-end developers.

**5.1.3 MYSQL**

We used MySQL as a database language for storing and communicating information with the database. It is domain specific language used in programming and designed for managing data held in relational database management system, or for stream processing in a relational data stream management system.

**5.1.4 XAMPP Server**

XAMPP is a free and open-source cross-platform web server solutions stack package developed by Apache stands for Cross-Platform(X), Apache(A), MariaDB(M), PHP(P), and Perl(P).We have used XAMPP server for the connection with the database.it is extremely easy for developer to create a local web server for testing and deployment purposes.

**5.2. Programming Tools**

**5.2.1 Front End Tools**

Front end also known as client-side is the practice of converting data to a graphical interface, through the use of React and CSS for a website or Web application, so that users can view and interact with that data directly. The objective of designing a site is to ensure that when the users open up the site they see the information in a format that is easy to read and relevant. They need to ensure that their site comes up correctly in different browsers, different operating system and different devices, which requires careful planning on the side of the developer.

* **React**

React makes it painless to create interactive UIs. Design simple views for each state in our application, and React will efficiently update and render just the right components when our data changes. Declarative views make our code more predictable and easier to debug.

* **Cascading Style Sheet (CSS)**

CSS is used to describe the presentation semantics (i.e. the look and formatting) of a document written in a markup language. It is designed primarily to enable the separation of document content (written in HTML or a similar markup language) from document presentation, including elements such as the layout, colors, and fonts.

**5.2.2 Back-End Tools**

The back-end of a website consists of a server, an application, and a database. A back-end developers builds and maintains the technology that powers those components which together, enable the user-facing side of the website to even exist in the first place. Simply back-end development is writing code that is not seen directly. A back-end is used to implementing algorithms and solving problems that front-end do and it is related to SQL and databases. It is also known as server-side scripting language consists of programming language like PHP, ASP.NET, JAVA, Ruby etc.

* **PHP**

PHP code may be embedded into html or html5 markup, or it can be used in combination with various web template systems, web content management systems, and web frameworks. PHP code is usually processed by a PHP interpreter implemented as a module in the web server or as a Common Gateway Interface (CGI) executable. The web server software combines the results of the interpreted and executed PHP code, which may be any type of data, including images, with the generated web page. PHP code may also be executed with a command-line interface (CLI) and can be used to implement standalone graphical applications.

1. PHP is a recursive acronym for “PHP: hypertext preprocessor”.
2. PHP is a server side scripting language that is embedded in html. It is used to manage dynamic content, databases, session tracking, even build entire e-commerce sites.
3. It is integrated with a number of popular databases, including MYSQL, Oracle, and Sybase, Informix, and Microsoft SQL server.

* **MySQL**

phpMyAdmin is used to create and manage the database in our system. Frequently used operations (managing databases, tables, columns, relations, indexes, users, permissions, etc.) can be performed via the user interface, while you still have the ability to directly execute any SQL statement.

**CHAPTER 6**

**IMPLEMENTATION AND TESTING**

**6.1. Implementation**

The implementation phase is less creative than system design. It is primarily concerned with user training, site preparation and file conversion. During file testing, user acceptance is, tested followed by user training. Depending on the nature of system, extensive user training may be required. Conversion usually takes place at about the same time the user is being trained. Programming is thought to be modified as a result of programming effort. Programming provides a “***realty test***” for the assumption made by analyst.

System testing checks the deadlines and accuracy of the system to access update and retrieve data from new files. Once the programs become available, test data are read into the computer and processed against the file provided. For testing, it successful program is then run with “***live***” data. Otherwise, a diagnostic test to locate and correct errors. In most conversion, parallel run is conducted where the new system simultaneous run both old system. This method through costly, provided added assurance against errors in the candidate system and also the user staff an opportunity to gain experience through operation.

**6.2. Testing**

Testing is an integral part of the software development process. It is performed at each stage of the SDLC. It ensures that the developed parts confirm to the user requirements. It helps to find out whether an input given to the system is well processed or not and output meets the specified objective of the system. It mainly ensures that the system performs as planned. The testing of system was carried out step by step. Testing is performed at key points that are crucial for the working of a system. In hardware system testing is done for individual components that are used to make up a system. In other word testing is a process of Validation and Verification. Validation is the process of checking if the system will meet the customer’s actual needs, whereas Verification is concerned whether the system is well-engineered and error free.

Various types of testing procedures were performed in order to check the working mechanism and correctness of the system. Some of the types of testing that we did attempt are described below:

**6.2.1 Unit Testing**

The purpose of unit testing in this project is to check that as individual parts of the system are functioning as expected or not. We have tested each module of this system in order to check the correctness of the output. Firstly, we found many errors and then it was resolved stepwise. For this testing, we test for order section by examining the orders and details provided by the users. We add order, in the database. Hence, unit testing is done.

**6.2.2 Integration Testing**

The purpose of integration test is to verify the functional, performance and reliability between the modules that are integrated.

**6.2.3 System Testing**

Prior to the actual implementation of the system it had to be tested comprehensively and every possible error uncovered. Since it is not possible to test the system exhaustively, the black box testing method was used for system testing. The black box testing usually demonstrates that software functions are operational, that the input is properly accepted and the output is correctly produced; and that integrity of external information (database) is maintained.

**6.2.4 Module Testing**

A module is composed of various programs related to that module. Module testing is done to check the module functionality and interaction between units within a module.

**6.2.5 Acceptance Testing**

Acceptance testing is done after implementation to check if the system runs successfully in the user’s environment.

**6.3. Result Analysis**

“Burger Ordering System” is a web based application which is still under development and even the functioning part are in various pieces. Various results have been tested for validation and verification process.

**6.3.1 Validation**

**Email Validation:**

When email address is submitted by user for subscription, it must follow the following expression otherwise it gets the error message.

/^\w+([.-]?\w+)\*@\w+([.-]?\w+)\*(\.\w{2,3})+$/

For example, [unique@bhuwan.com](mailto:unique@bhuwan.com) is a valid email address but [unique@bhuwan@fashion.com](mailto:unique@bhuwan@fashion.com) is an invalid email address.

**CHAPTER 7**

**CONCLUSION AND RECOMMENDATION**

**7.1. Achievement of the Project**

This project has gone through a series of activities to develop a complex solution for the online Burger Ordering System. After analysis of the project’s goal and research direction, a set of objectives were established. All the activities done during the project were attempts to realize these objectives.

**7.2. Limitations of the System**

There are some of the limitations of the system. The user are only able to add the provided ingredients in their burger. The payment method is only cash-on-delivery, so it limits the paying system. Requires skilled technical personnel for CRUD (create, read, update, delete) operations of database which may not always be possible.

**7.3. Future Recommendation**

In addition to the unfinished requirements, there are other possibilities of further improving the project. The improvements may include:

* Overcoming the limitations mentioned above.
* May add more payment systems.
* Can add more ingredients to add in the burger.
* Adding the support to tracking location of the delivery.
* Can add the price of ingredients manually.

**7.4. Conclusion**

Hence, we have developed web-based system named as “Burger Ordering System”. Our project allows user to select their ingredients for their burger. The quantity of the ingredients added determines the actual price of the burger. User can also view all the ingredients and the price of each ingredients. The status of the ordered burger can also be viewed by the user.

**APPENDIX I (SNAPS SHOTS)**

1. **Login Page (Back End)**

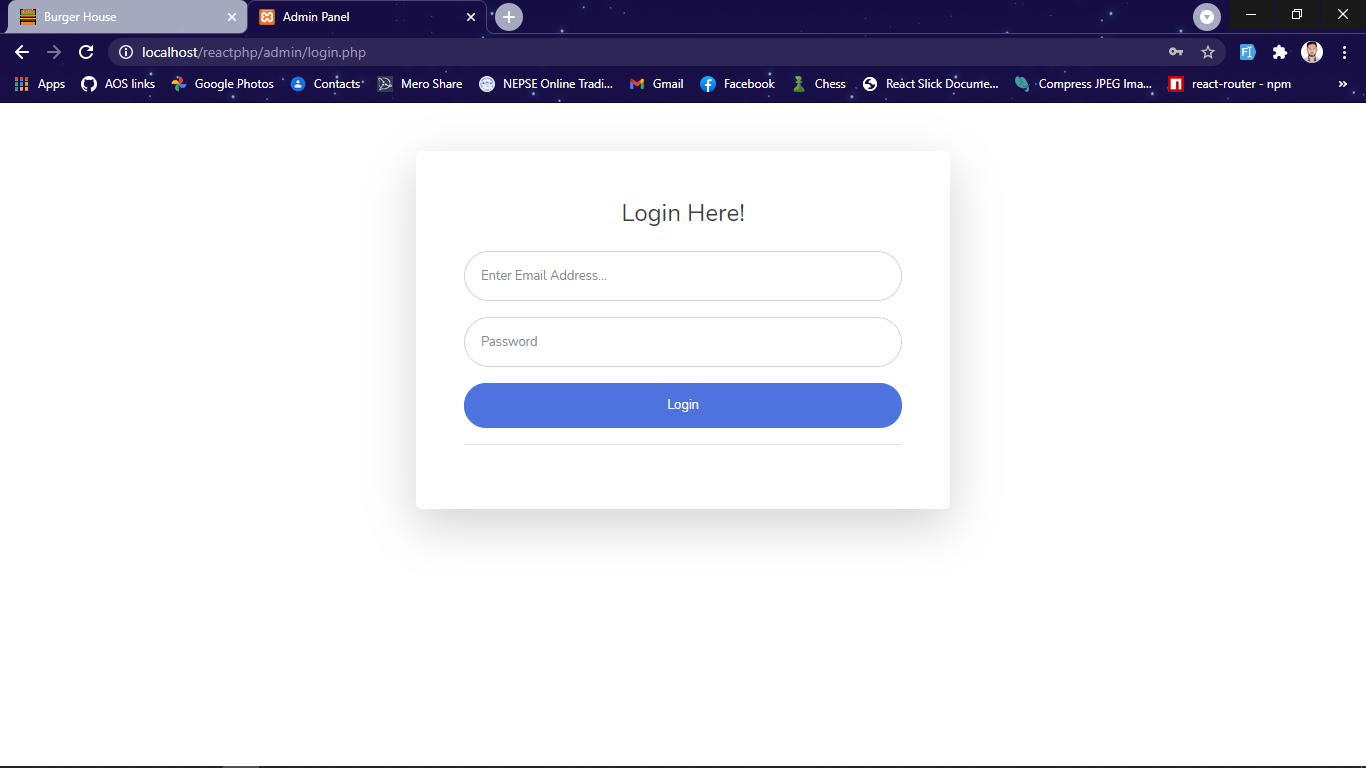
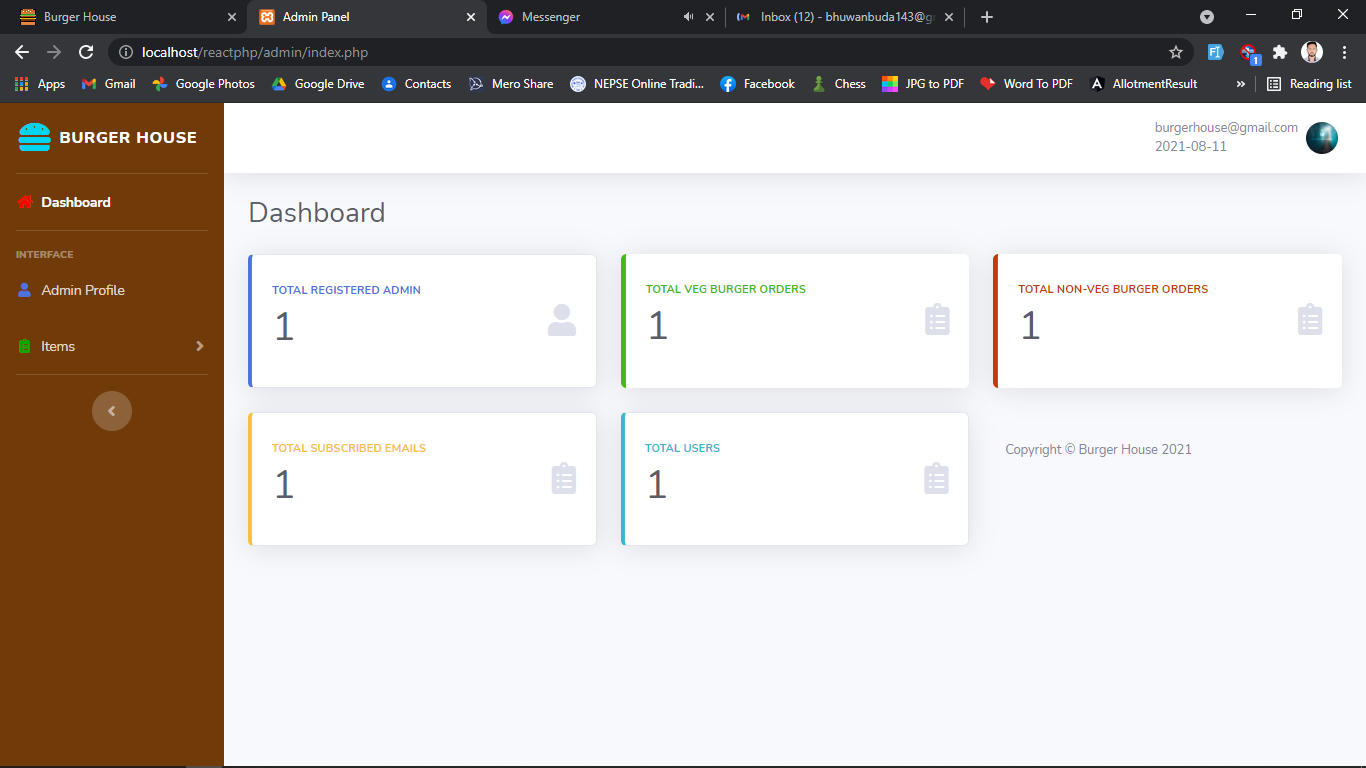


Figure 1: Login Page

**2. Home Page (Back End)**

******

*Figure 2: Home Page*

**3. View Admin Page (Back End)**

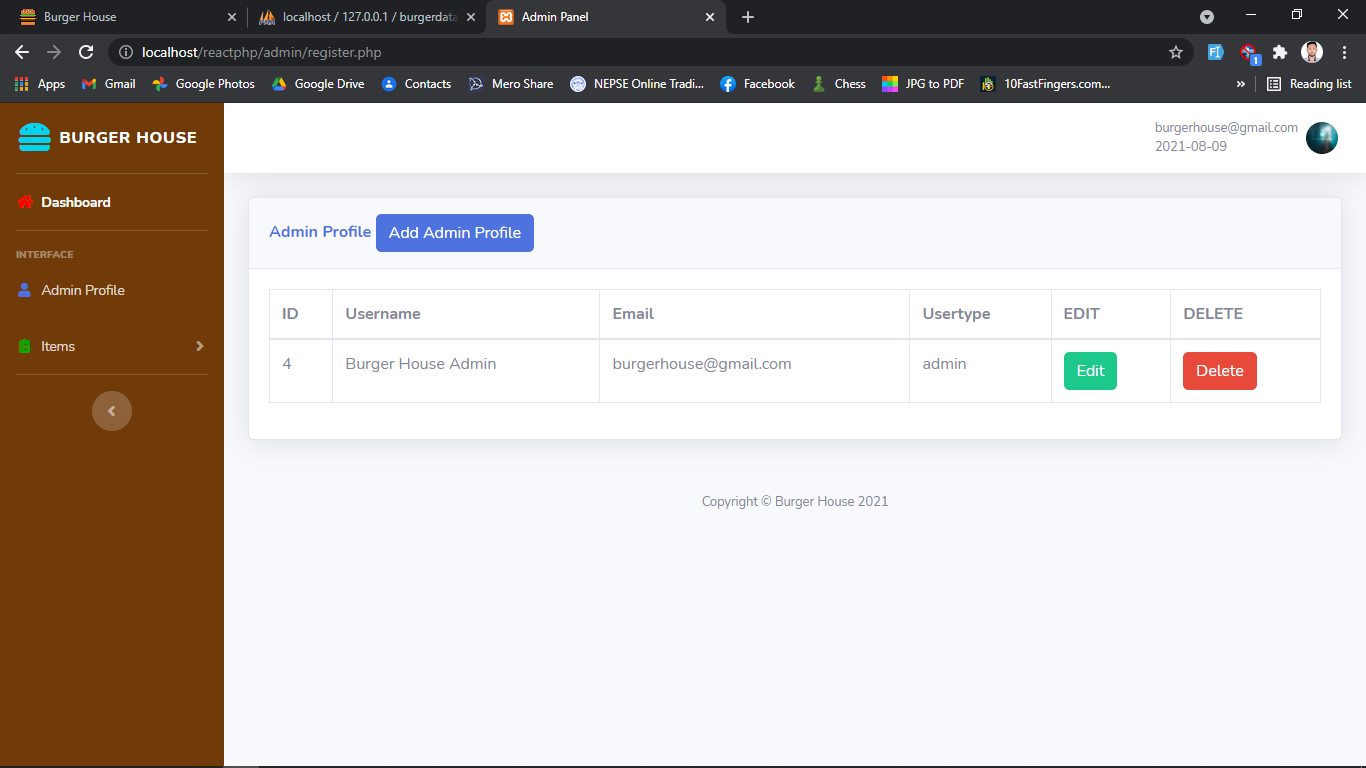


Figure 3: View Admin Page

**4. Add Admin page (Back End)**

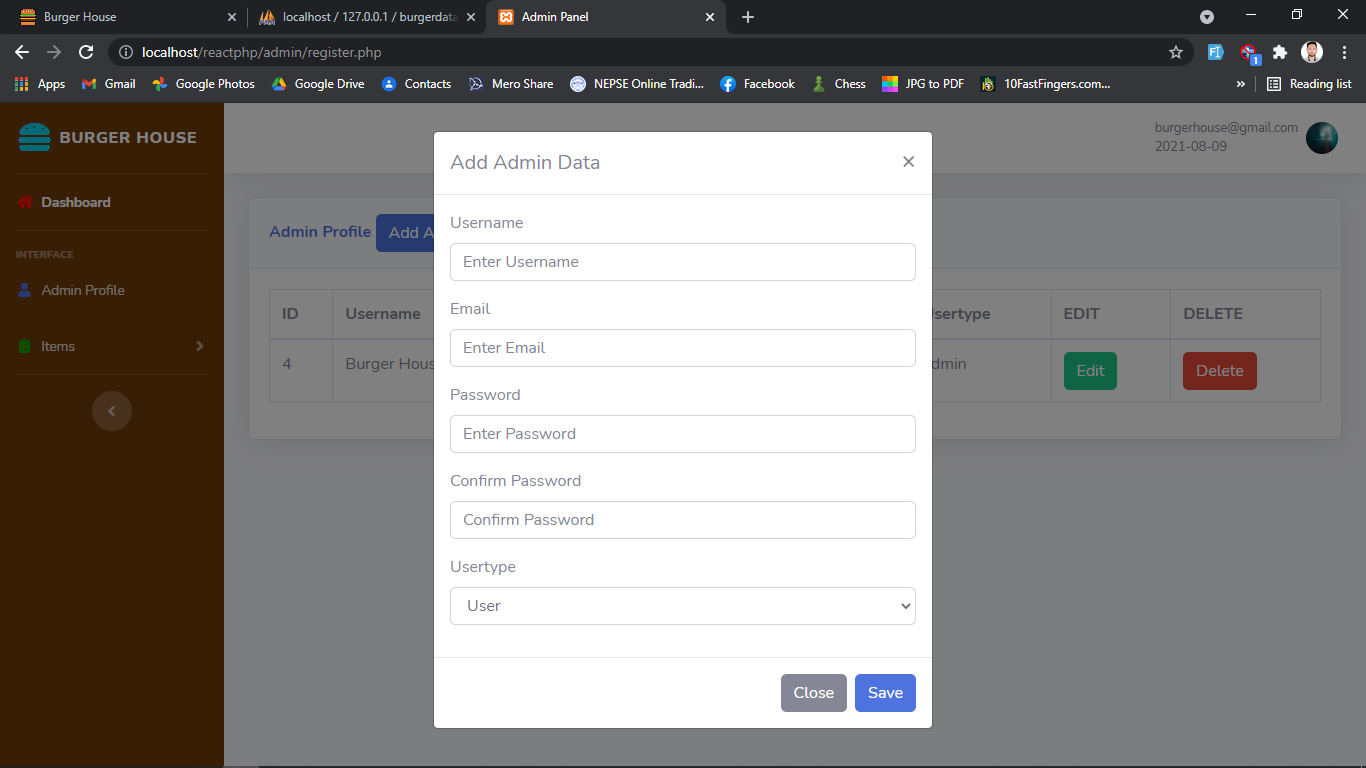


Figure 4: Add Admin Page

**5. View Veg Order Page (Back End)**

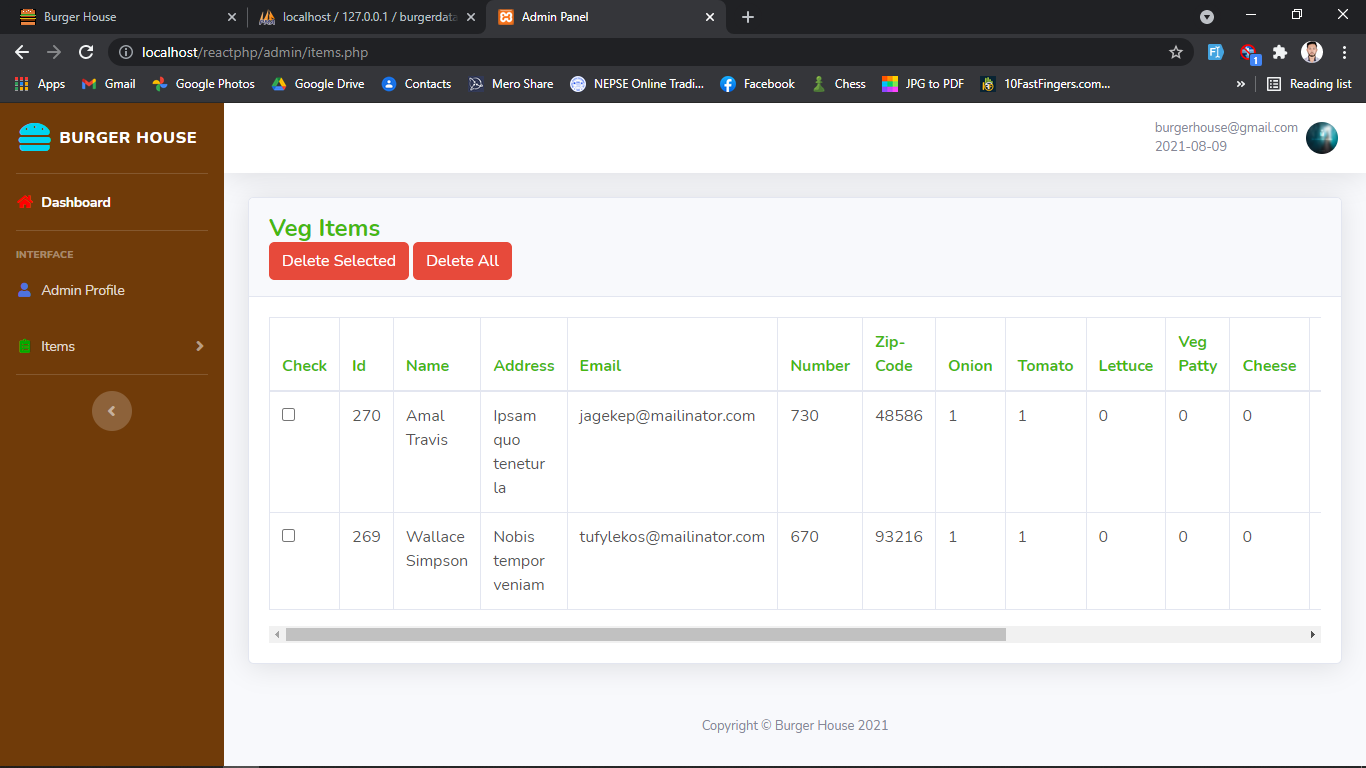
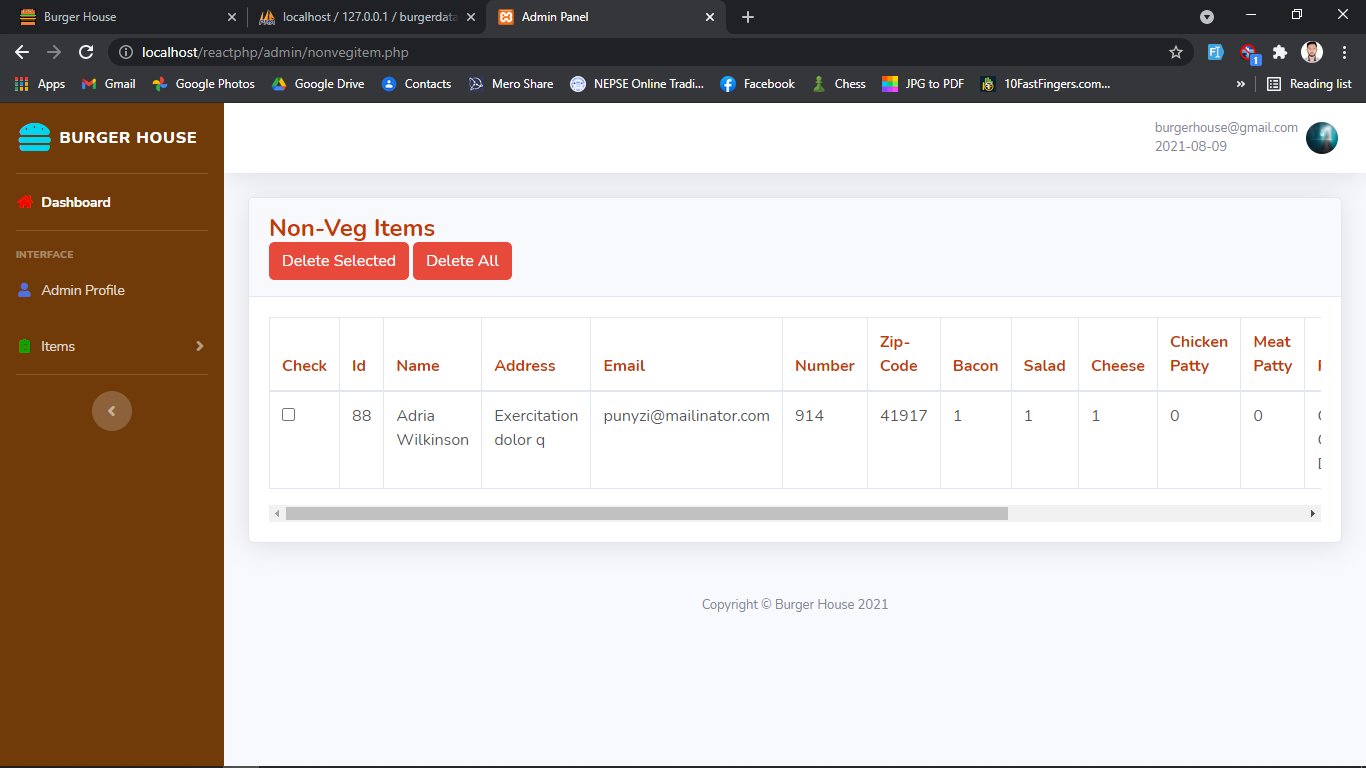


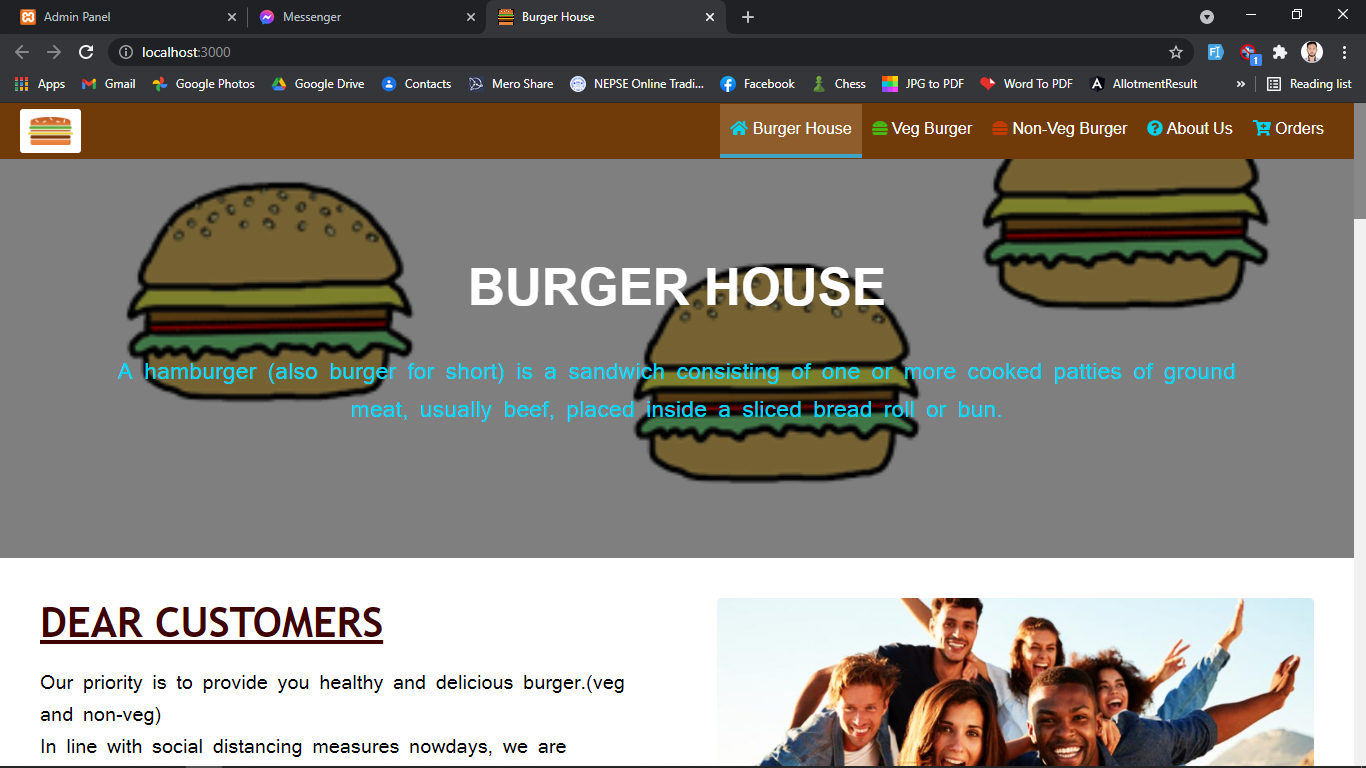
Figure 5: View Veg Order Page

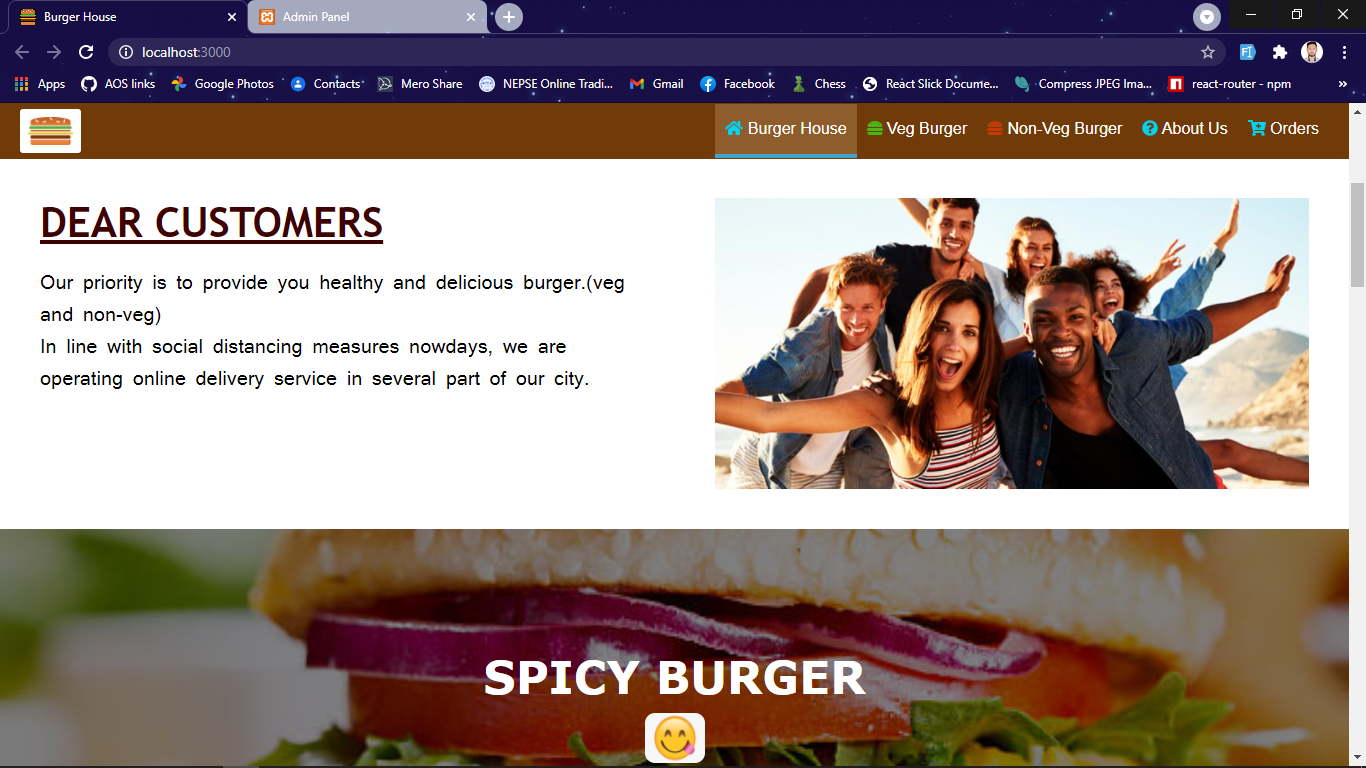
**6. View Non-Veg Order Page (Back End)**

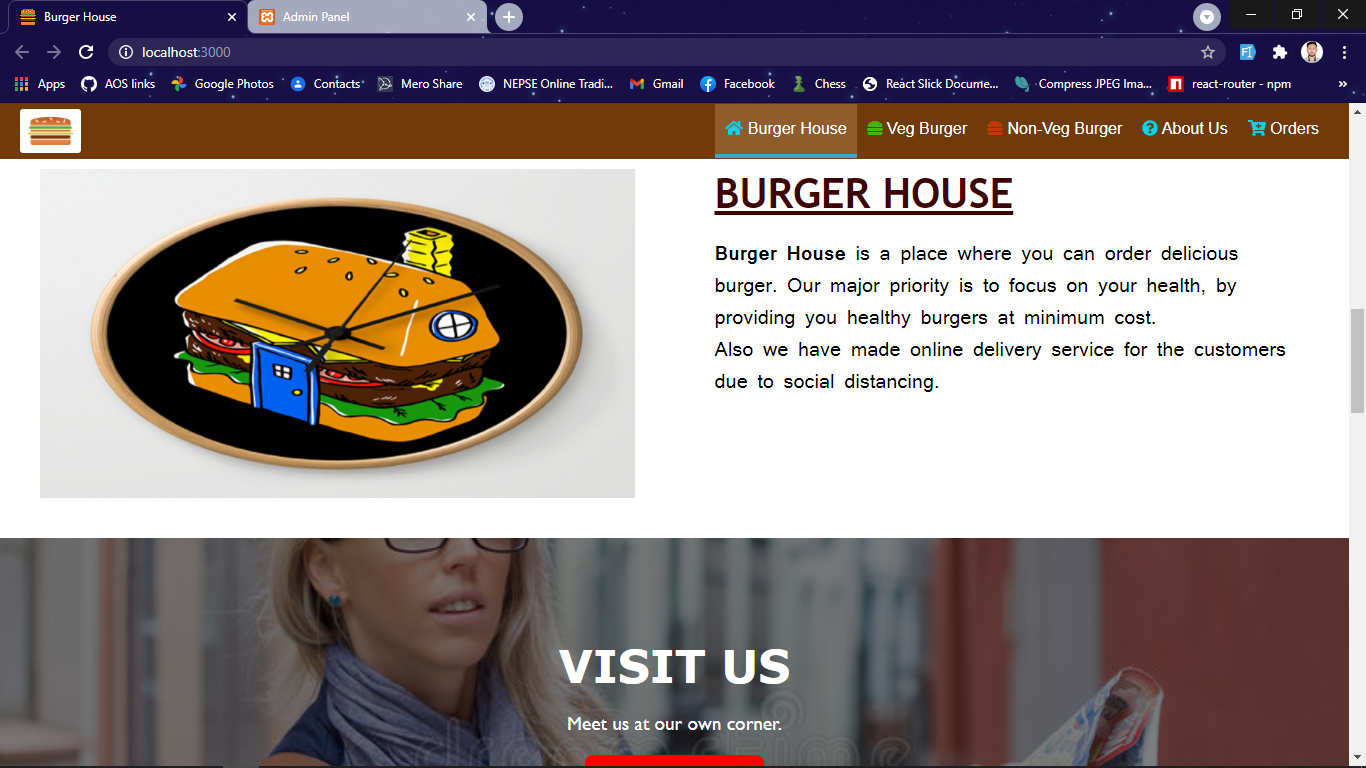
****

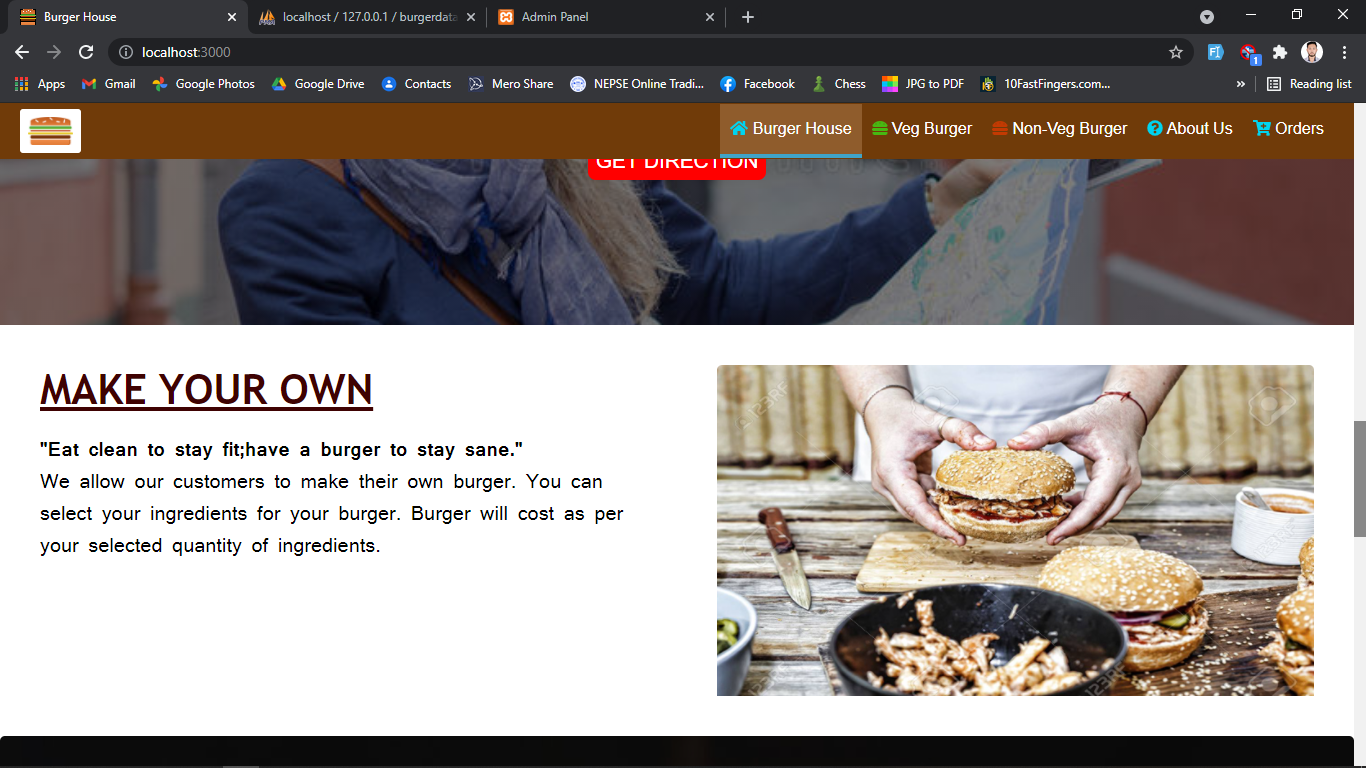
*Figure 6: View Non-Veg Order Page*

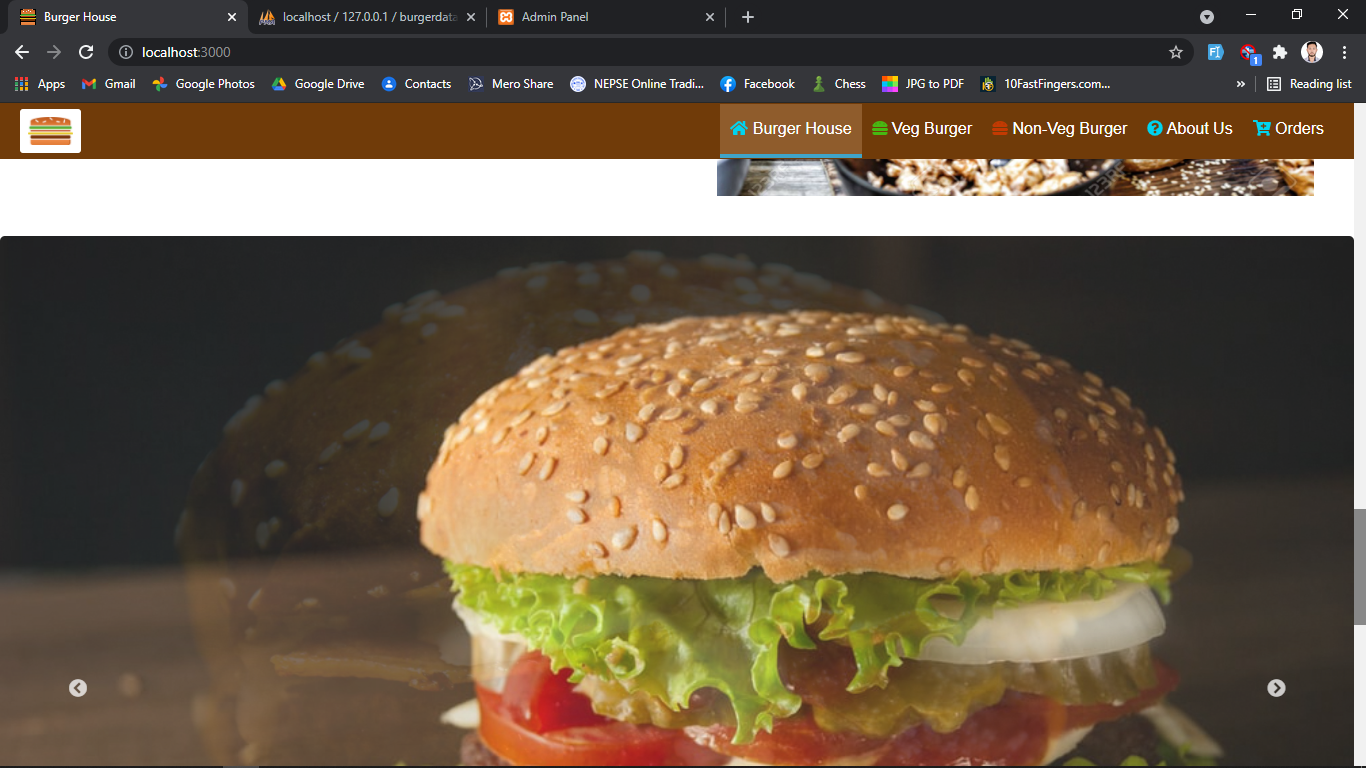
**7. Homepage (Front End)**

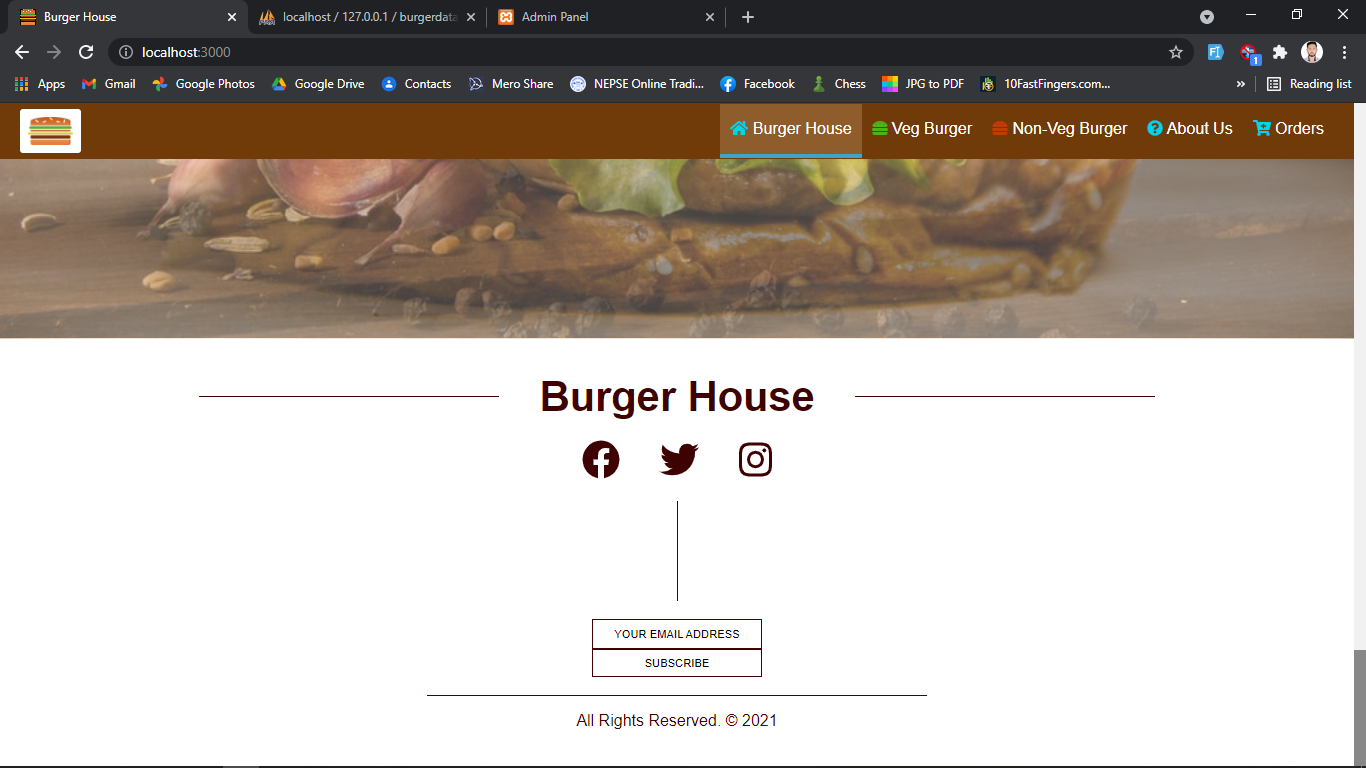
****



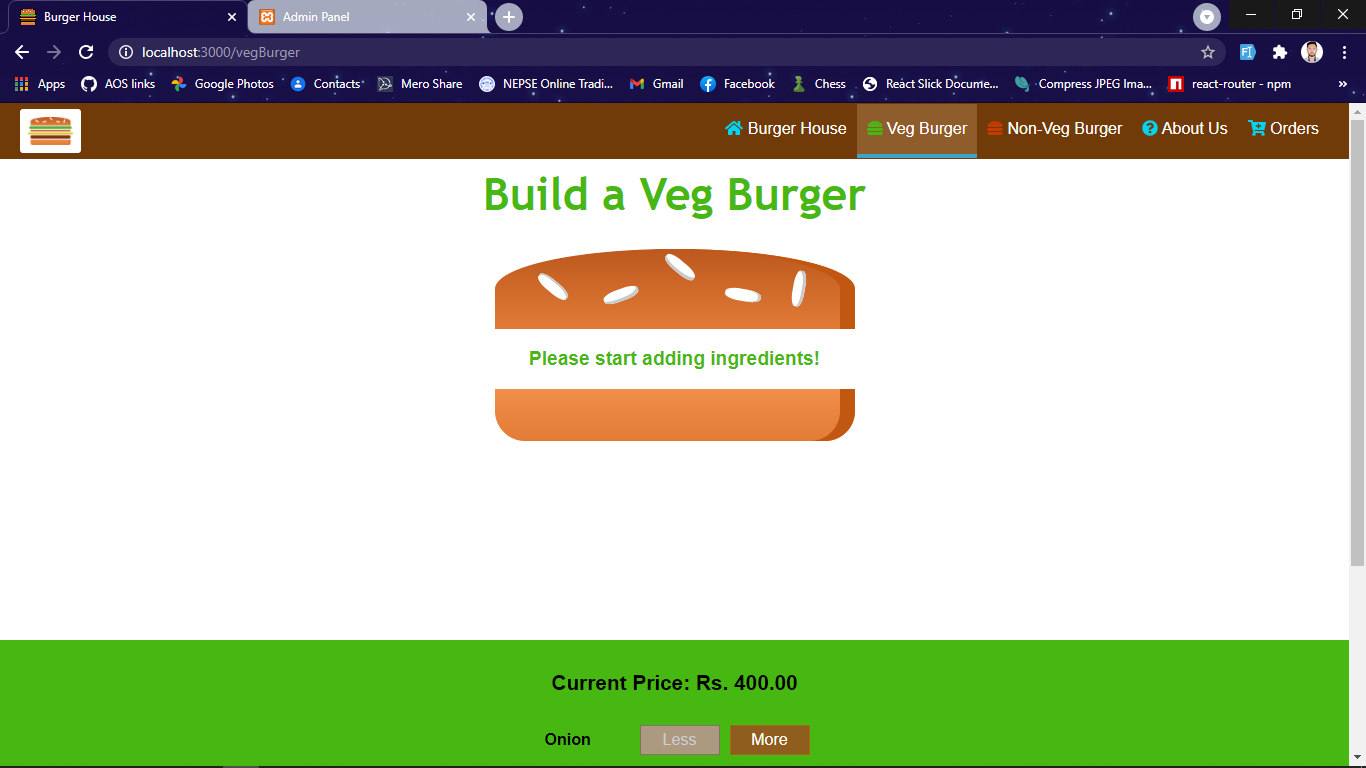
****

****

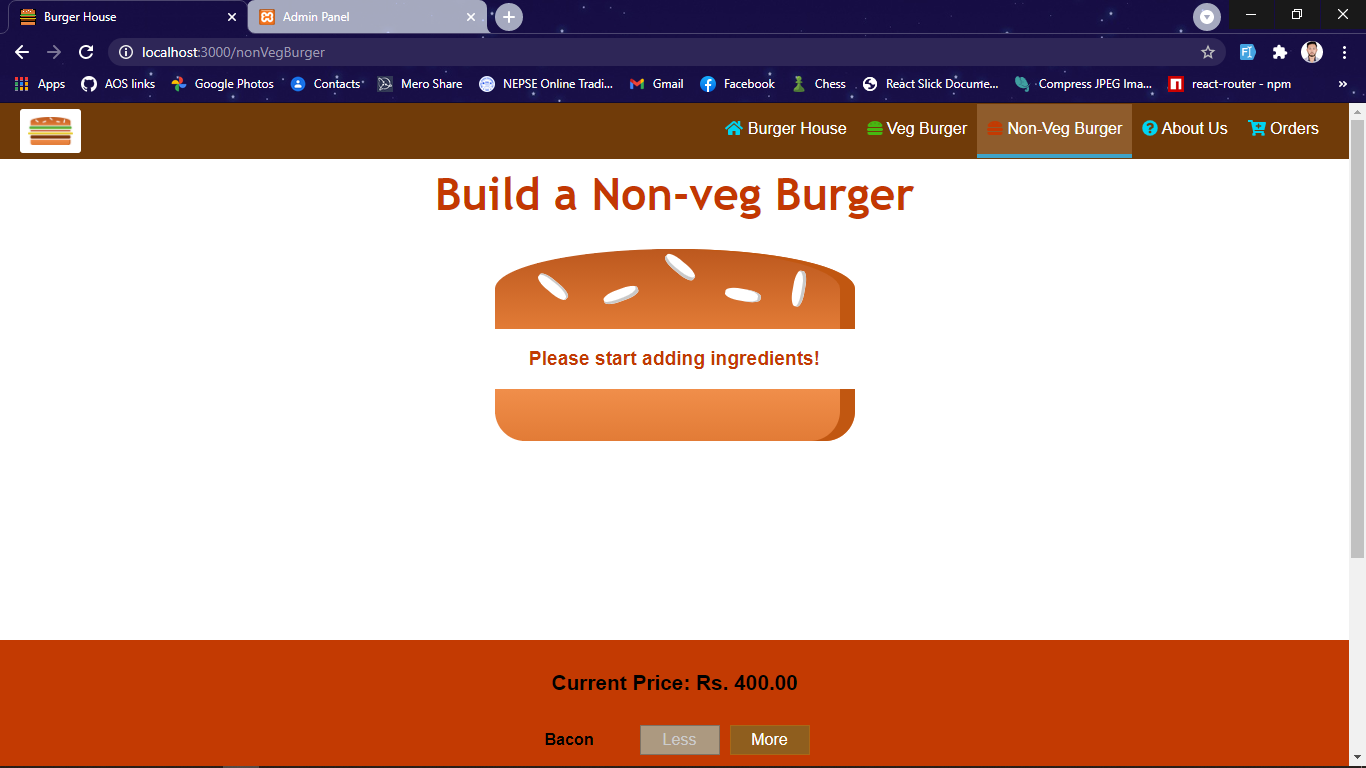
****

****

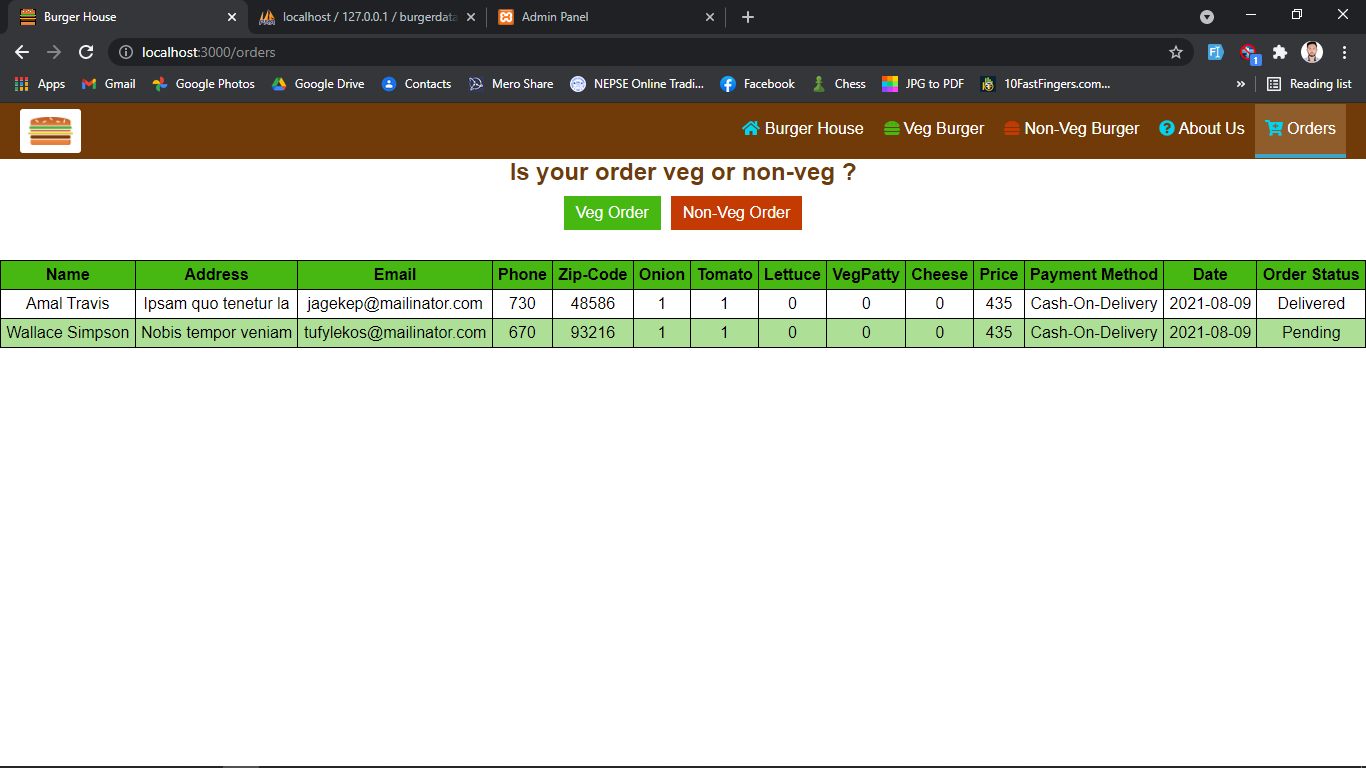
**8. Veg Burger Page (Front End)**

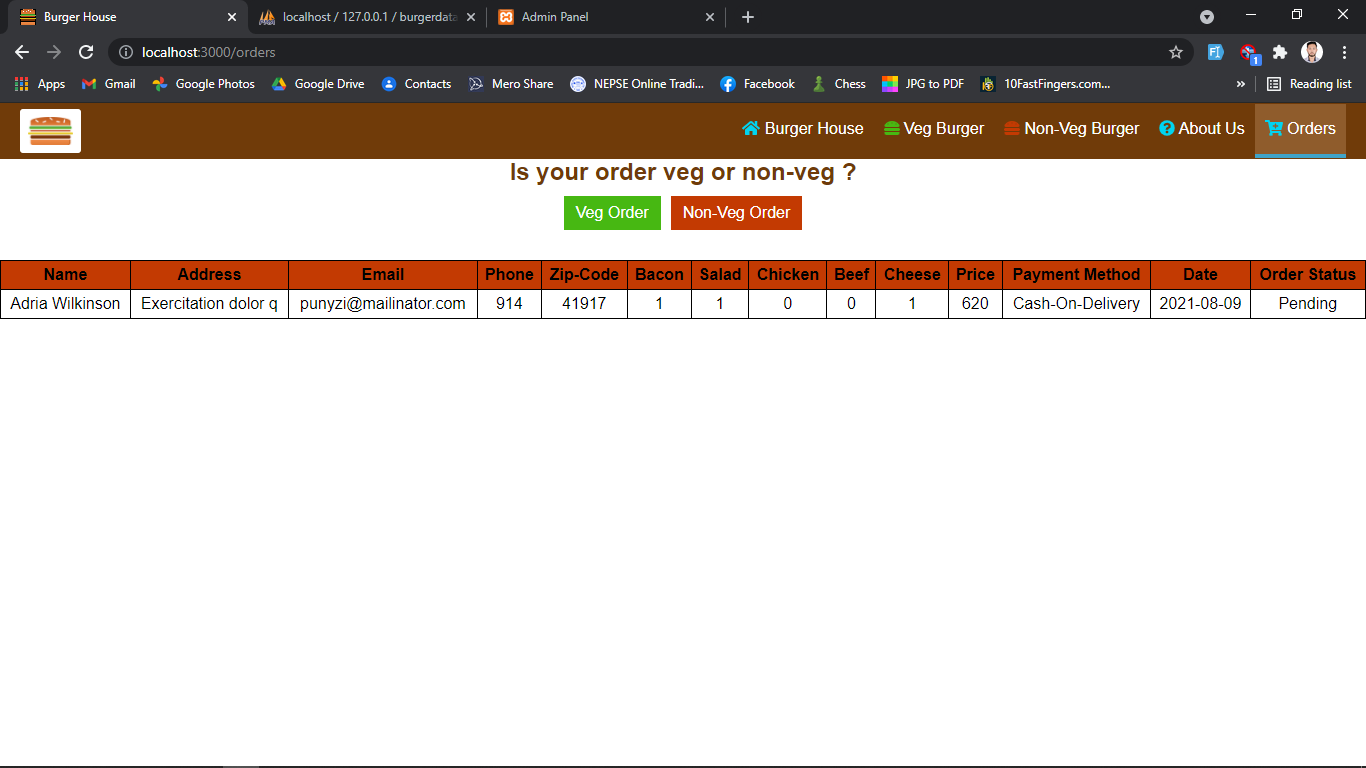
****

**9. Non-Veg Burger Page (Front End)**

****

**10. Orders Page (Front End)**

****

****

**REFERENCES**

**Our References:**

[1] https://reactjs.org/

[2] https://en.wikipedia.org/wiki/Hamburger

[3] https://www.youtube.com/channel/UCpX0Eh1tiNj02w6of3ZFeFg

[4] <https://www.stackoverflow.com>

[5] https://www.w3schools.com/php/default.asp