**FarmFolio**

*Mini Project Report*

*Submitted by*

**RohithR Nair**

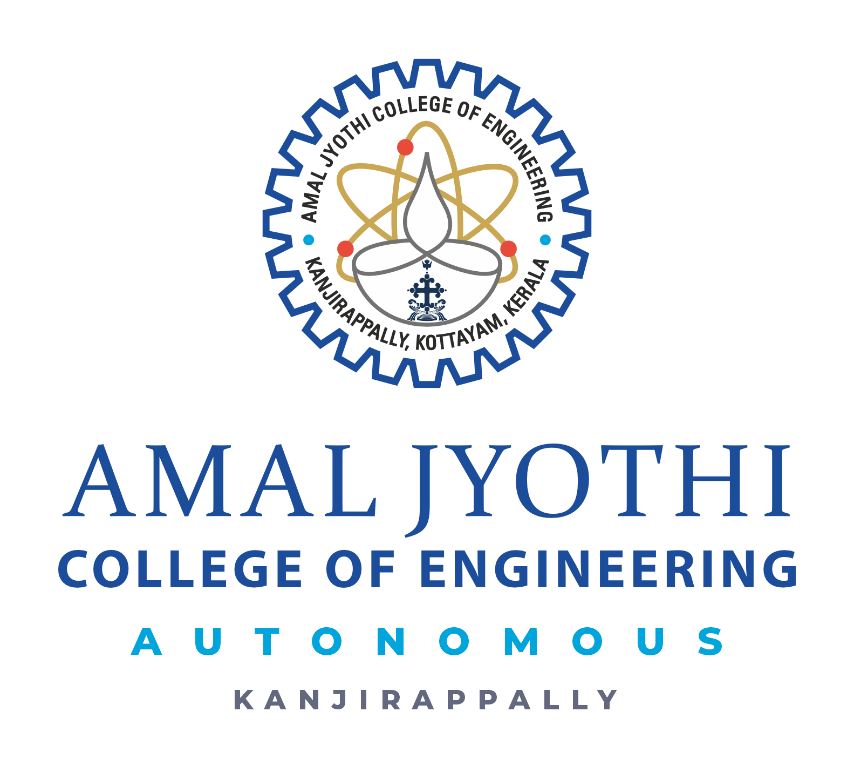
**Reg. No.: AJC22MCA-I050**

*In Partial fulfillment for the Award of the Degree of*

**INTEGRATED MASTER OF COMPUTER APPLICATIONS**

**(INMCA)**

**APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY**



**AMAL JYOTHI COLLEGE OF ENGINEERING AUTONOMOUS**

**KANJIRAPPALLY**

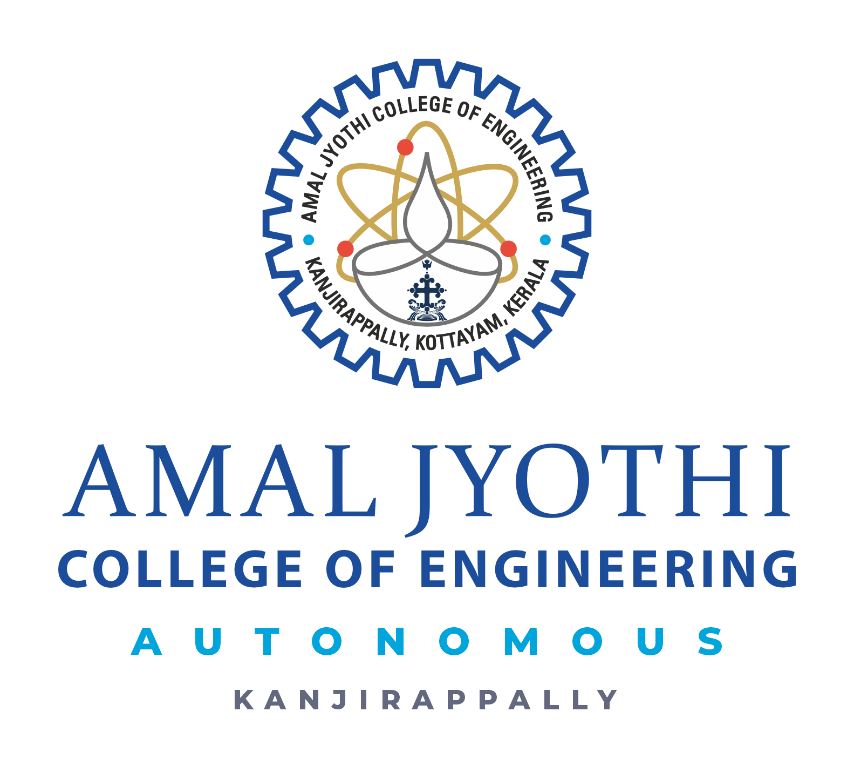
[Affiliated to APJ Abdul Kalam Technological University, Kerala. Approved by AICTE, Accredited by NAAC. Koovappally, Kanjirappally, Kottayam, Kerala – 686518]

# 2024-2025

## DEPARTMENT OF COMPUTER APPLICATIONS

### AMAL JYOTHI COLLEGE OF ENGINEERING AUTONOMOUS

**KANJIRAPPALLY**



**CERTIFICATE**

This is to certify that the Project report, “**FARMFOLIO”** is the bona fide work of **ROHITH R NAIR (Regno: AJC22MCA-I050)** carried out in partial fulfillment of the requirements for the award of the **Degree of Integrated Master of Computer Applications** at **Amal Jyothi College of Engineering Autonomous, Kanjirappally,** Affiliated to **APJ Abdul Kalam Technological University**. The project was undertaken during the period from **January 01, 2025 to April 23, 2025.**

**AJITH GS MEERA ROSE MATHEW**

**Internal Guide Coordinator**

**Rev. Fr. Dr. Rubin Thottupurathu Jose**

**Head of the Department**

**DECLARATION**

I hereby declare that the project report **“FARMFOLIO”** is a bona fide work done at **Amal Jyothi College of Engineering Autonomous, Kanjirappally**, Affiliated to **APJ Abdul Kalam Technological University**, towards the partial fulfilment of the requirements for the award of the **Integrated Master of Computer Applications (INMCA)** during the period from **January 01, 2025 to April 23, 2025.**

**Date: ROHITH R NAIR**

**KANJIRAPPALLY Reg: AJC22MCA-I050**

# ACKNOWLEDGEMENT

First and foremost, I thank God almighty for his eternal love and protection throughout the project. I take this opportunity to express my gratitude to all who helped me in completing this project successfully. It has been said that gratitude is the memory of the heart. I wish to express my sincere gratitude to our Director (Administration) **Rev. Fr. Dr. Roy Abraham Pazhayaparampil** and Principal **Dr. Lillykutty Jacob** for providing good faculty for guidance.

I owe a great depth of gratitude towards our Head of the Department **Rev.Fr.Dr. Rubin Thottupurathu Jose** for helping us. I extend my whole hearted thanks to the project coordinator **MEERA Rose Mathew** for her valuable suggestions and for overwhelming concern and guidance from the beginning to the end of the project. I would also express sincere gratitude to my guide **Ajith GS** for his inspiration and helping hand.

I thank our beloved teachers for their cooperation and suggestions that helped me throughout the project. I express my thanks to all my friends and classmates for their interest, dedication, and encouragement shown towards the project. I convey my hearty thanks to my family for the moral support, suggestions, and encouragement to make this venture a success.

ROHITH R NAIR

# ABSTRACT

Farmfolio is an innovative web platform designed to bridge the gap between farm owners and consumers by creating a direct and user-friendly interface. The platform allows farm owners to list their farms, provide details about their products, and share contact information, enabling consumers to connect with them directly. Unlike conventional marketplaces, Farmfolio emphasizes simplicity and transparency, excluding chat functionality while providing direct phone numbers for quick and efficient communication.

**Four types of users:**

* Farm Owners – Who can register their farms, list available products, and update their offerings.
* Consumers – Who can browse through farms, contact owners, and access fresh products.
* Delivery Boys - Who are responsible for delivering the products ordered by consumers from the farms. They can receive delivery requests, track orders, and update the status of deliveries.
* Admin – Responsible for managing platform activities and ensuring smooth operations.

**Farmfolio also introduces two standout features:**

• Farm Ratings and Reviews: Empower consumers to rate and review farms, helping others make informed choices and fostering trust.

• Farm Events: Enable farm owners to promote activities such as farm visits, workshops, and local markets, enriching the consumer experience and supporting community engagement.

The platform is developed using HTML, CSS, JavaScript, PHP and MYSQL ensuring a robust and responsive interface. By streamlining the connection between farms and consumers, Farmfolio aims to support sustainable agricultural practices, promote local businesses, and provide fresh, high-quality items to communities.

**CONTENT**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SL. NO** | | **TOPIC** | **PAGE NO** | |
| **1** | | **INTRODUCTION** |  | |
| **1.1** | | **PROJECT OVERVIEW** |  | |
| **1.2** | | **PROJECT SPECIFICATION** |  | |
| **2** | | **SYSTEM STUDY** |  | |
| **2.1** | | **INTRODUCTION** |  | |
| **2.2** | | **EXISTING SYSTEM** |  | |
| **2.3** | | **DRAWBACKS OF EXISTING SYSTEM** |  | |
| **2.4** | | **PROPOSED SYSTEM** |  | |
| **2.5** | | **ADVANTAGES OF PROPOSED SYSTEM** |  | |
| **3** | | **REQUIREMENT ANALYSIS** |  | |
| **3.1** | | **FEASIBILITY STUDY** |  | |
| **3.1.1** | | **ECONOMICAL FEASIBILITY** |  | |
| **3.1.2** | | **TECHNICAL FEASIBILITY** |  | |
| **3.1.3** | | **BEHAVIORAL FEASIBILITY** |  | |
| **3.1.4** | | **FEASIBILITY STUDY QUESTIONNAIRE** |  | |
| **3.2** | | **SYSTEM SPECIFICATION** |  | |
| **3.2.1** | | **HARDWARE SPECIFICATION** |  | |
| **3.2.2** | | **SOFTWARE SPECIFICATION** |  | |
| **3.3** | | **SOFTWARE DESCRIPTION** |  | |
| **3.3.1** | | **PHP** |  | |
| **3.3.2** | | **MYSQL** |  | |
| **4** | | **SYSTEM DESIGN** |  | |
| **4.1** | | **INTRODUCTION** |  | |
| **4.2** | | **UML DIAGRAM** |  | |
| **4.2.1** | | **USE CASE DIAGRAM** |  | |
| **4.2.2** | | **SEQUENCE DIAGRAM** |  | |
| **4.2.3** | | **STATE CHART DIAGRAM** |  | |
| **4.2.4** | | **ACTIVITY DIAGRAM** |  | |
| **4.2.5** | | **CLASS DIAGRAM** |  | |
| **4.2.6** | | **OBJECT DIAGRAM** |  | |
| **4.2.7** | | **COMPONENT DIAGRAM** |  | |
| **4.2.8** | | **DEPLOYMENT DIAGRAM** |  | |
| **4.2.9** | | **COLLABORATION DIAGRAM** |  | |
| **4.3** | | **USER INTERFACE DESIGN USING FIGMA** |  | |
| **4.4** | | **DATABASE DESIGN** |  | |
| **5** | | **SYSTEM TESTING** |  | |
| **5.1** | | **INTRODUCTION** |  | |
| **5.2** | | **TEST PLAN** |  | |
| **5.2.1** | **UNIT TESTING** | |  |
| **5.2.2** | **INTEGRATION TESTING** | |  |
| **5.2.3** | **VALIDATION TESTING** | |  |
| **5.2.4** | **USER ACCEPTANCE TESTING** | |  |
| **5.2.5** | **AUTOMATION TESTING** | |  |
| **5.2.6** | **SELENIUM TESTING** | |  |
| **6** | **IMPLEMENTATION** | |  |
| **6.1** | **INTRODUCTION** | |  |
| **6.2** | **IMPLEMENTATION PROCEDURE** | |  |
| **6.2.1** | **USER TRAINING** | |  |
| **6.2.2** | **TRAINING ON APPLICATION SOFTWARE** | |  |
| **6.2.3** | **SYSTEM MAINTENANCE** | |  |
| **7** | **CONCLUSION & FUTURE SCOPE** | |  |
| **7.1** | **CONCLUSION** | |  |
| **7.2** | **FUTURE SCOPE** | |  |
| **8** | **BIBLIOGRAPHY** | |  |
| **9** | **APPENDIX** | |  |
| **9.1** | **SAMPLE CODE** | |  |
| **9.2** | **SCREEN SHOTS** | |  |

## List of Abbreviations

# CHAPTER 1

# INTRODUCTION

### PROJECT OVERVIEW

### Farmfolio is a web platform designed to connect farm owners directly with consumers, enabling them to browse farms, access fresh products, and contact owners without intermediaries. The platform features four user roles—Farm Owners, Consumers, Delivery Personnel, and Admin—each with specific functionalities to ensure seamless operations. Key features include farm ratings and reviews for trust-building and farm event promotions to enhance community engagement. Built using HTML, CSS, JavaScript, PHP, and MySQL, Farmfolio streamlines farm-to-consumer interactions, supports local agriculture, and promotes sustainable farming practices.

### PROJECT SPECIFICATION

* **Product & Farm Management** – Farm owners can register, list products, update availability, and promote farm events, while consumers can browse and connect directly.
* **Order & Delivery System** – Delivery personnel receive and track delivery requests, ensuring efficient order fulfillment and real-time status updates.
* **Ratings & Reviews** – Consumers can rate and review farms to foster trust and help others make informed choices.
* **Technology Stack** – Developed using HTML, CSS, JavaScript for the frontend, PHP for backend logic, and MySQL for database management, ensuring a robust and scalable system.

# CHAPTER 2

# SYSTEM STUDY

# 

### INTRODUCTION

Farmfolio is a web-based platform designed to bridge the gap between farm owners and consumers by providing a direct and efficient marketplace for fresh farm products. The platform eliminates intermediaries, allowing consumers to connect with farm owners and purchase directly. Additionally, it includes features for delivery management, farm event promotions, and farm ratings, ensuring a streamlined experience.

### EXISTING SYSTEM

### Currently, farm owners rely on traditional marketplaces, social media, or third-party e-commerce platforms to sell their products. These methods often involve high commission fees, lack of transparency, and limited consumer engagement. Additionally, communication between buyers and farm owners is inefficient, leading to delays and reduced trust.

**2.2.1 NATURAL SYSTEM STUDIED**

The natural system observed is the conventional farm-to-consumer model, where buyers visit local farms, farmers’ markets, or rely on middlemen to purchase fresh produce. This system, while organic and community-driven, lacks scalability, convenience, and accessibility for a broader audience.

**2.2.2 DESIGNED SYSTEM STUDIED**

Several existing online agricultural marketplaces were studied, including e-commerce platforms that allow farmers to list their products. However, most of these systems involve complex interfaces, commission-based transactions, or lack direct communication between buyers and farm owners. Additionally, few platforms focus on farm events and local engagement.

### DRAWBACKS OF EXISTING SYSTEM

 High dependency on middlemen, increasing costs for consumers.

 Lack of a centralized, user-friendly platform for farm owners to manage sales and events.

 Limited trust-building mechanisms such as verified ratings and reviews.

 Inefficient order fulfillment and delivery tracking systems.

 Poor direct communication between consumers and farm owners.

### PROPOSED SYSTEM

Farmfolio aims to resolve these issues by introducing a user-friendly platform where farm owners can list their farms and products, and consumers can browse and contact them directly. Delivery personnel ensure efficient order fulfillment, and the platform supports farm event promotions, enhancing consumer engagement. The system eliminates chat functionality in favor of direct contact via phone numbers for quick communication.

### ADVANTAGES OF PROPOSED SYSTEM

* **Direct Farm-to-Consumer Connection** – Eliminates intermediaries, reducing costs and ensuring fresh produce.
* **Enhanced Trust & Transparency** – Features like farm ratings and reviews build credibility.
* **Efficient Order & Delivery System** – Dedicated delivery personnel streamline logistics.
* **Community Engagement** – Farm event promotions encourage local participation and awareness.
* **User-Friendly & Scalable** – Simple interface built with HTML, CSS, JavaScript, PHP, and MySQL for scalability and ease of use.

# CHAPTER 3

# REQUIREMENT ANALYSIS

## FEASIBILITY STUDY

### Economical Feasibility

### Technical Feasibility

### Behavioral Feasibility

**3.1.4 Feasibility Study Questionnaire**

## SYSTEM SPECIFICATION

### Hardware Specification

Processor -

RAM -

Hard disk -

### Software Specification

Front End -

Back End -

Database -

Client on PC - Windows 7 and above.

Technologies used - JS, HTML5, AJAX, J Query, PHP, CSS, List APIs

## SOFTWARE DESCRIPTION

### Eg.PHP

### Eg. MySQL

# CHAPTER 4

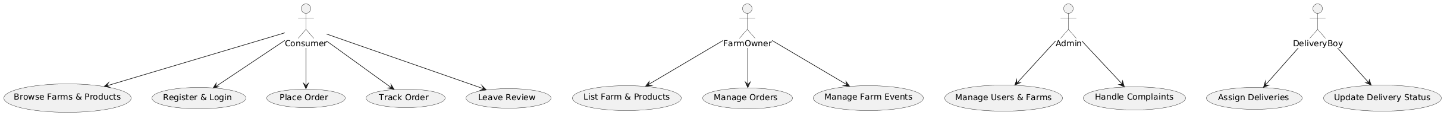
# SYSTEM DESIGN

* 1. **INTRODUCTION**

## UML DIAGRAM

## USE CASE DIAGRAM

Explanation, Diagram



**1. Consumer**

* **Browse Farms & Products** – Consumers can explore available farms and their listed products.
* **Register & Login** – Required for accessing personalized features and placing orders.
* **Place Order** – Consumers can order farm products directly.
* **Track Order** – Check the status of their placed orders.
* **Leave Review** – Provide feedback and ratings for farms to enhance trust and credibility.

**2. Farm Owner**

* **Manage Orders** – View and process orders placed by consumers.
* **Manage Farm Events** – Create and manage events such as farm visits or workshops to engage consumers.

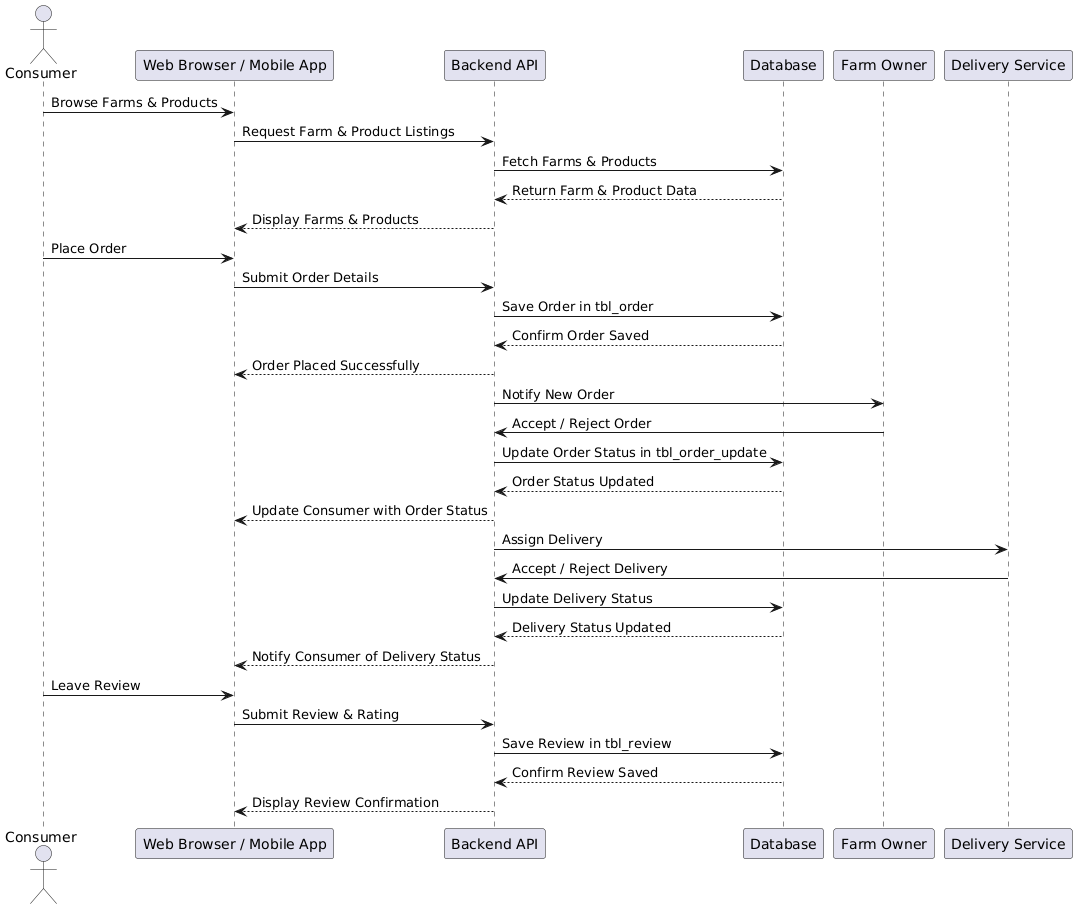
**3. Admin**

* **Manage Users & Farms** – Oversee user registrations, farm listings, and platform activities.
* **Handle Complaints** – Address issues and complaints from users to maintain smooth platform operations.

**4. Delivery Boy**

* **Assign Deliveries** – Receive and manage delivery requests.
* **Update Delivery Status** – Mark deliveries as completed or update their progress.

## SEQUENCE DIAGRAM

Explanation, Diagram

**1. Browsing Farms & Products**

* The **Consumer** initiates a request to browse farm listings via the **Web Browser/Mobile App**.
* The request is sent to the **Backend API**, which fetches farm data from the **Database**.
* The retrieved data is returned and displayed to the consumer.

**2. Placing an Order**

* The **Consumer** submits order details via the **Web Browser/Mobile App**.
* The **Backend API** saves the order in the **Database** and confirms successful placement.
* The **Farm Owner** is notified of the new order and can accept or reject it.
* Once the order status is updated, the consumer is notified.

**3. Order Delivery Process**

* The **Farm Owner** assigns the order for delivery, notifying the **Delivery Service**.
* The **Delivery Service** can accept or reject the request.
* The **Delivery Status** is updated in the **Database**, and the consumer is notified.

**4. Leaving a Review**

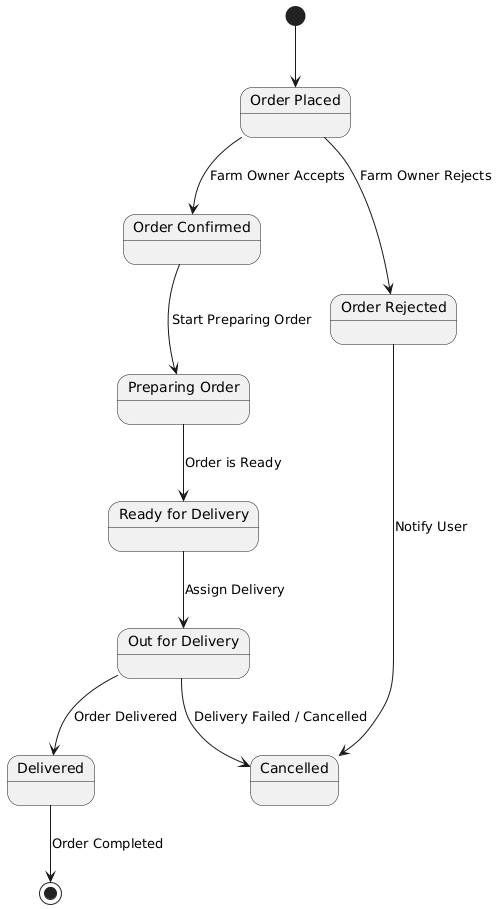
* The **Consumer** submits a review and rating through the **Web Browser/Mobile App**.
* The **Backend API** saves the review in the **Database**, and a confirmation is displayed to the user.

**Key Takeaways**

* The diagram effectively maps out the **step-by-step interactions** between the **Consumer, Backend API, Database, Farm Owner, and Delivery Service**.
* The **order placement and delivery tracking system** ensures transparency and efficiency.
* The **review system** helps build trust within the platform.

## 4.2.2 State Chart Diagram

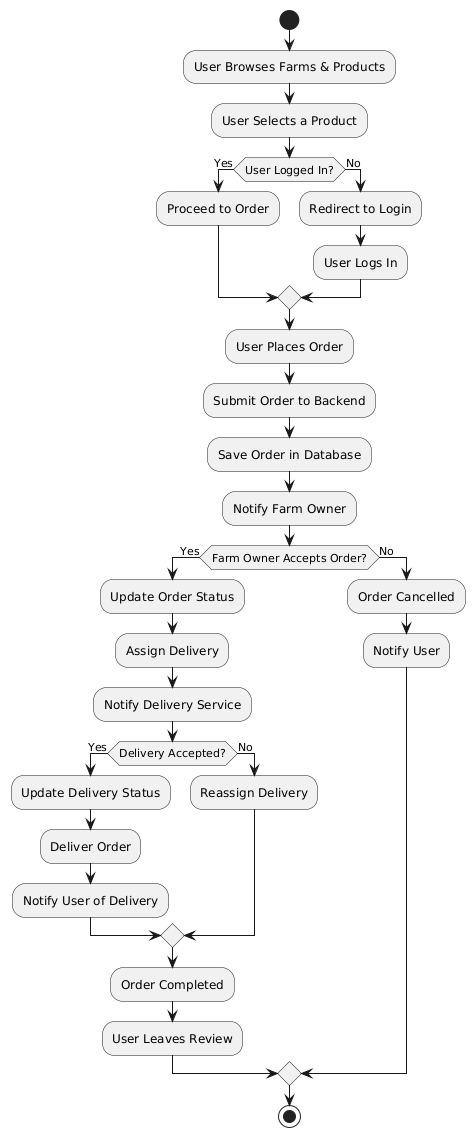
Explanation, Diagram



**Flow Explanation:**

1. **Order Placed** – The process starts when a consumer places an order.
2. **Order Decision** – The **Farm Owner** can either **accept** or **reject** the order:
   * If **accepted**, the order moves to **Order Confirmed**.
   * If **rejected**, the user is notified, and the process ends.
3. **Order Preparation** – Once confirmed, the farm owner starts preparing the order.
4. **Ready for Delivery** – After preparation, the order is marked as **ready**, and delivery is assigned.
5. **Out for Delivery** – The delivery personnel picks up the order and starts the delivery process.
6. **Final Order Status**:
   * If successfully delivered, the status is marked as **Delivered**, and the process is **completed**.
   * If there is a failure or cancellation, the order is marked as **Cancelled**, and the process ends.

## Activity Diagram

Explanation, Diagram

 **User Browses Farms & Products** – The consumer explores available farms and their offerings.

 **Product Selection & Login Check** – If the user selects a product but is not logged in, they are redirected to log in before proceeding with the order.

 **Placing the Order** – Once logged in, the user places the order, which is sent to the backend and stored in the database. The farm owner is notified.

 **Farm Owner Decision** –

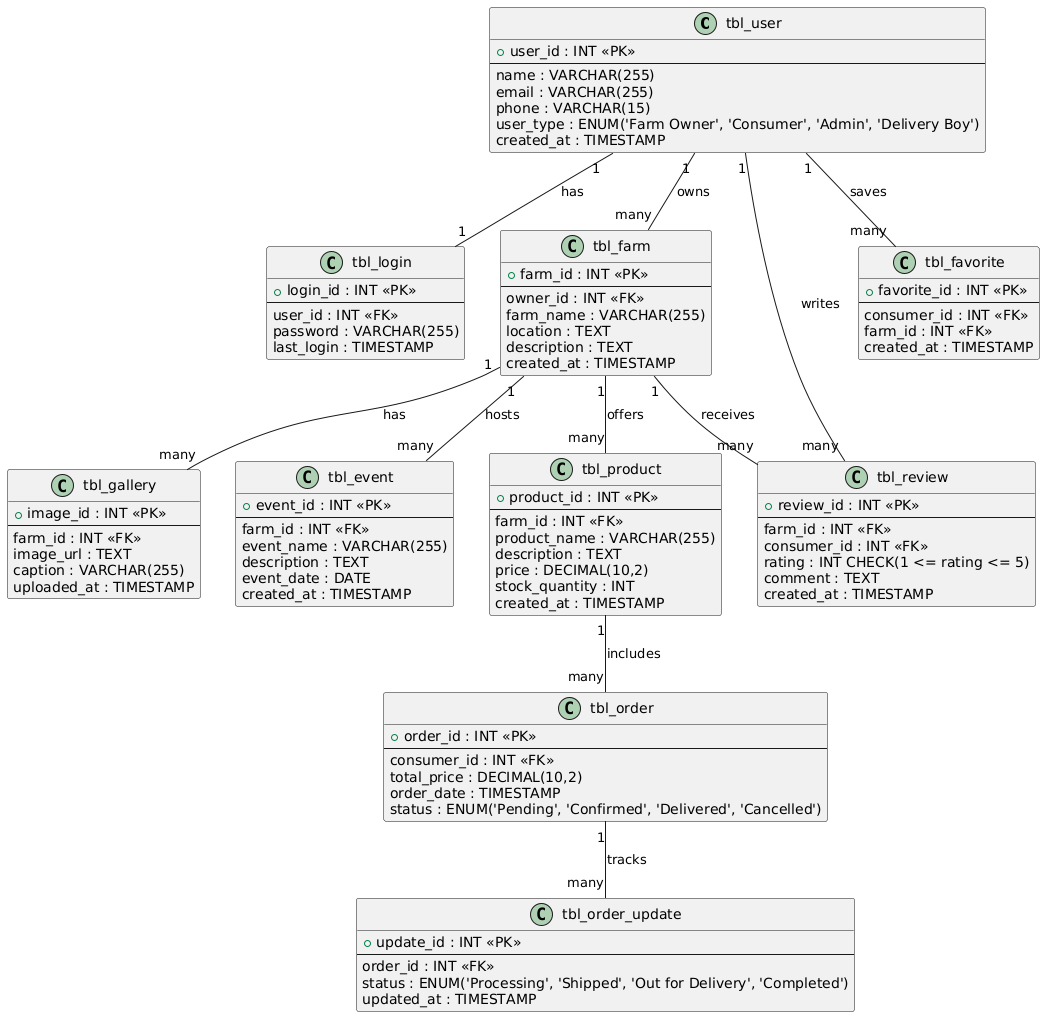
* If the **order is accepted**, the status is updated, and delivery is assigned.
* If the **order is rejected**, it is canceled, and the user is notified.

 **Delivery Assignment** – The delivery service is notified.

* If the **delivery is accepted**, the status is updated, and the order is delivered.
* If the **delivery is rejected**, it is reassigned.

 **Order Completion & Review** – Once the order is successfully delivered, the user is notified and can leave a review.

## Class Diagram

Explanation, Diagram

 **tbl\_user**

* Stores user details such as name, email, phone, and type (e.g., Farm Owner, Consumer, Admin, Delivery Boy).
* Linked to **tbl\_login** for authentication.

 **tbl\_farm**

* Represents farms owned by users (Farm Owners).
* Contains details like farm name, location, and description.
* A farm **hosts many** events (**tbl\_event**) and **offers many** products (**tbl\_product**).

 **tbl\_product**

* Stores farm products, including name, description, price, and stock quantity.
* A farm **offers many** products.

 **tbl\_order**

* Tracks consumer purchases, including total price, order date, and status (Pending, Confirmed, Delivered, Cancelled).
* An order **includes multiple** products and is tracked through **tbl\_order\_update**.

 **tbl\_order\_update**

* Logs order status updates (Processing, Shipped, Out for Delivery, Completed).

 **tbl\_review**

* Consumers can leave **ratings (1-5) and comments** for farms.

 **tbl\_favorite**

* Consumers can **save farms** to their favorites for quick access.

 **tbl\_gallery**

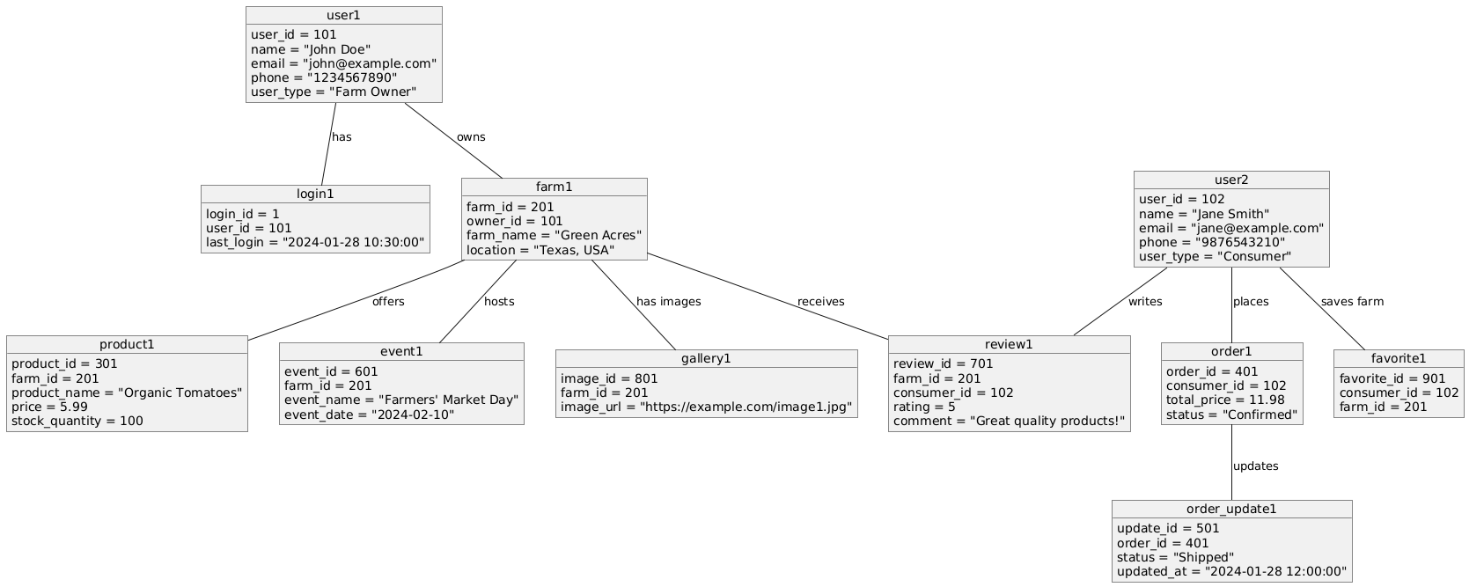
* Stores images related to farms.

 **tbl\_event**

* Represents farm-hosted events with details like name, description, and event date.

## Object Diagram

Explanation, Diagram



 **Users**

* **user1 (John Doe)** is a **Farm Owner** who owns **farm1 (Green Acres)**.
* **user2 (Jane Smith)** is a **Consumer** who places orders and writes reviews.
* Users have login credentials stored in **login1**.

 **Farm (farm1)**

* Owned by **user1 (John Doe)**.
* Offers **product1 (Organic Tomatoes)**.
* Hosts **event1 (Farmers' Market Day)**.
* Has images stored in **gallery1**.

 **Products (product1)**

* **Organic Tomatoes** with a price of **$5.99** and a stock quantity of **100**.

 **Orders (order1)**

* **user2 (Jane Smith)** placed an order (order\_id = 401) for **farm1’s product**.
* The total price is **$11.98**, and the status is **Confirmed**.
* Order status is tracked in **order\_update1**, currently marked as **Shipped**.

 **Reviews (review1)**

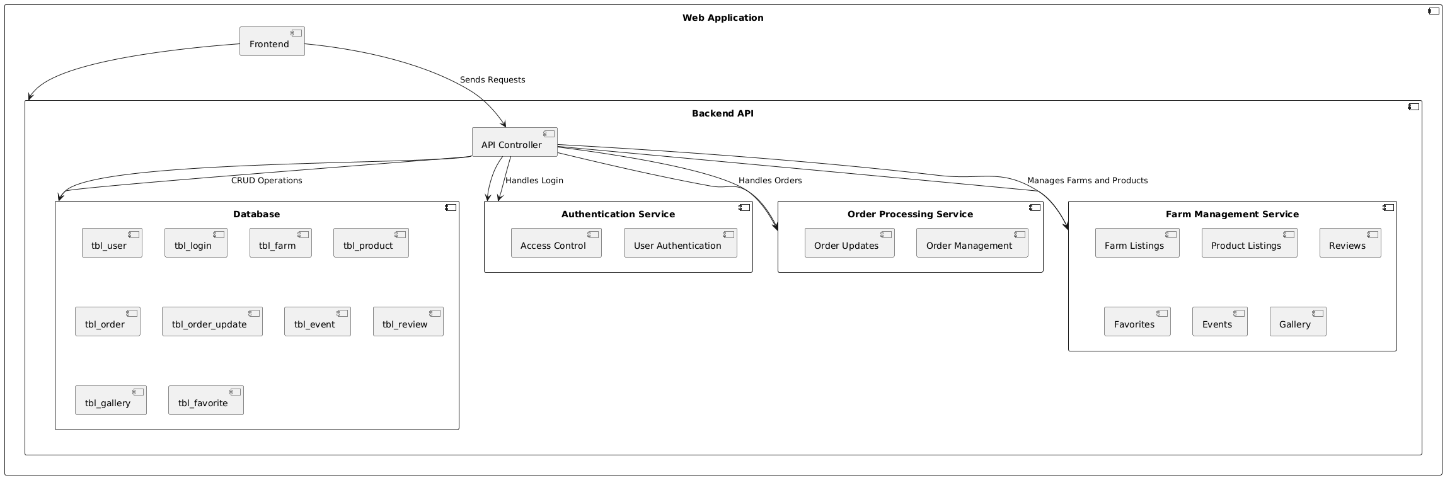
* **user2 (Jane Smith)** rated **farm1** with **5 stars** and commented, "Great quality products!"

 **Favorites (favorite1)**

* **user2 (Jane Smith)** saved **farm1** to their favorite farms list.

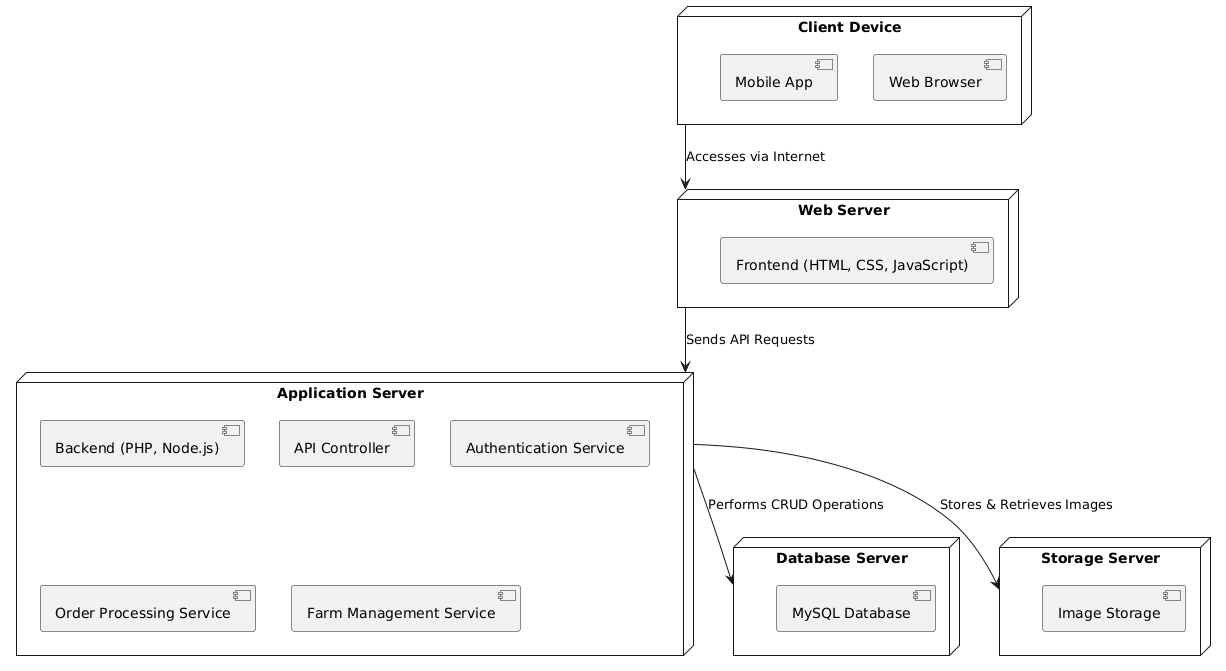
## Component Diagram

Explanation, Diagram



1. **Web Application (Frontend)**
   * The user interface where consumers, farm owners, and admins interact.
   * Sends requests to the **Backend API** for authentication, order processing, and farm management.
2. **Backend API**
   * Acts as the central controller, handling requests from the frontend and interacting with different services.
   * Includes an **API Controller** responsible for routing requests.
3. **Database**
   * Stores essential information, including users (tbl\_user, tbl\_login), farms (tbl\_farm, tbl\_product), orders (tbl\_order, tbl\_order\_update), events (tbl\_event), and reviews (tbl\_review).
4. **Authentication Service**
   * Manages **user authentication and access control**, ensuring only authorized users can access specific functionalities.
5. **Order Processing Service**
   * Handles **order management and updates**, ensuring smooth transactions between consumers and farm owners.
6. **Farm Management Service**
   * Manages **farm listings, product listings, reviews, favorites, events, and gallery** for better consumer-farm interaction.

**4.2.8 Deployment Diagram**

Explanation, Diagram

 **Client Device**

* Users access the platform via a **Mobile App** or **Web Browser** over the internet.

 **Web Server**

* Hosts the **frontend**, developed using **HTML, CSS, and JavaScript**.
* Sends API requests to the **Application Server** for processing.

 **Application Server**

* Manages the core functionalities of the platform.
* Includes:
  + **Backend (PHP, Node.js)** – Handles business logic and processes requests.
  + **API Controller** – Routes requests between different services.
  + **Authentication Service** – Manages user login and security.
  + **Order Processing Service** – Handles order placements and updates.
  + **Farm Management Service** – Manages farm listings, products, and reviews.

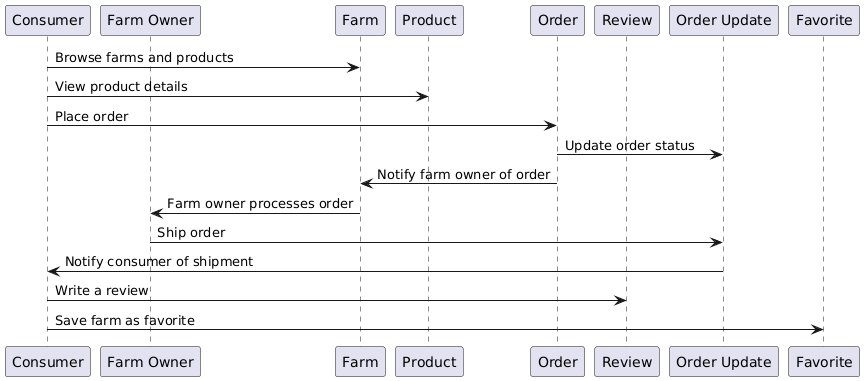
 **Database Server**

* Uses **MySQL** to store and manage data, including users, farms, orders, and reviews.
* Performs **CRUD (Create, Read, Update, Delete) operations** for the system.

 **Storage Server**

* Stores and retrieves **images** related to farms, products, and events.

**4.2.9 Collaboration Diagram**

Explanation, Diagram

 **Consumer Actions:**

* Browses farms and products.
* Views product details.
* Places an order.

 **Farm and Order Processing:**

* The system notifies the farm owner of the order.
* The farm owner processes and ships the order.
* The system updates the order status.
* The consumer is notified about the shipment.

 **Post-Purchase Actions:**

* The consumer writes a review.
* The consumer saves the farm as a favorite.

## 4.3 USER INTERFACE DESIGN USING FIGMA

**Form Name: abcc**

Screenshot

**Form Name: abcc**

Screenshot

All Forms

## 4.4 DATABASE DESIGN

### 4.4.1 Relational Database Management System (RDBMS)

Explanation

### 4.4.2 Normalization

Explanation with example

### 4.4.3 Sanitization

**4.4.4 Indexing**

### 4.5 TABLE DESIGN

### 1.tbl\_signup

Primary key: **userid**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No:** | **Field name** | **Datatype (Size)** | **Key Constraints** | **Description of the field** |
| 1 | userid | Int(11) | PRIMARY KEY NOT NULL | Unique identifier for each user |
| 2 | username | varchar(50) | NOT NULL | Username for user |
| 3 | mobile | varchar(11) | NOT NULL | Mobike numbem |
| 4 | email | varchar(100) | NOT NULL | Email |
| 5 | house | varchar(255) | NOT NULL | House name |
| 6 | district | varchar(100) | NOT NULL | District |
| 7 | state | varchar(100) | NOT NULL | State |
| 8 | pin | Char(6) | NOT NULL | Pin |
| 9 | password | Varchar(255) | NOT NULL | Password |
| 10 | Signup\_time | datetime | NOT NULL | Store the signup date and time |

**2.tbl\_login**

Primary key: **login\_id**

Foreign key: **userid** references table **tbl\_sighup**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Datatype (Size)** | **Key Constraints** | **Description of the Field** |
| 1 | login\_id | int(11) | Primary Key, Not Null | Unique identifier for each login entry |
| 2 | email | varchar(255) | Not Null | User's email address for login |
| 3 | password | varchar(255) | Not Null | Encrypted password for authentication |
| 4 | type | int(11) | Not Null | User type (e.g., Admin, Consumer) |
| 5 | login\_time | timestamp | Not Null, Default current\_timestamp() | Timestamp of login entry |
| 6 | userid | int(11) | Foreign Key, Not Null | Reference to the user in tbl\_user |
| 7 | username | varchar(50) | Not Null | Username associated with the user |

**3.tbl\_farms**

Primary key: **farm\_id**

Foreign key: **user\_id** references table **tbl\_signup**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Datatype (Size)** | **Key Constraints** | **Description of the Field** |
| 1 | farm\_id | int(11) | Primary Key, Not Null | Unique identifier for each farm |
| 2 | user\_id | int(11) | Foreign Key, Not Null | References the owner of the farm (from tbl\_signup) |
| 3 | farm\_name | varchar(255) | Not Null | Name of the farm |
| 4 | location | varchar(255) | Not Null | Address or geographical location of the farm |
| 5 | description | text | Default NULL | Additional details about the farm |
| 6 | created\_at | timestamp | Not Null, Default current\_timestamp() | Timestamp of when the farm was added |
| 7 | status | enum('pending','active','rejected') | Not Null, Default 'pending' | Current status of the farm (pending, active, or rejected) |

**4.tbl\_farms**

Primary key: **category\_id**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Datatype (Size)** | **Key Constraints** | **Description of the Field** |
| 1 | category\_id | int(11) | Primary Key, Not Null | Unique identifier for each category |
| 2 | category | varchar(255) | Not Null | Name of the main category |
| 3 | sub | varchar(100) | Not Null | Name of the subcategory |
| 4 | status | enum('0','1') | Not Null, Default '1' | Status of the category (0 = inactive, 1 = active) |

# CHAPTER 5

# SYSTEM TESTING

* 1. **INTRODUCTION**

Explanation

## TEST PLAN

Explanation

### Unit Testing

explanation

### Integration Testing

Explanation

### Validation Testing or System Testing

Explanation

### Output Testing or User Acceptance Testing

explanation.

* + 1. **Automation Testing**

explanation.

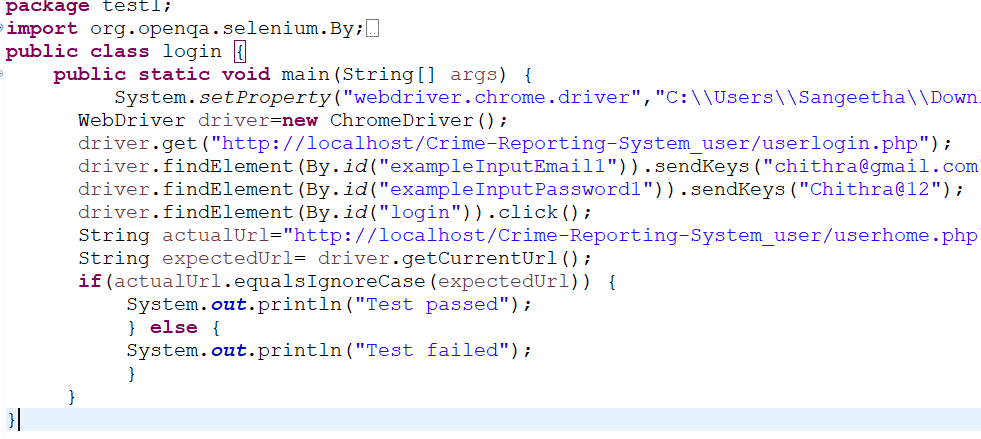
* + 1. **Selenium Testing**

explanation.

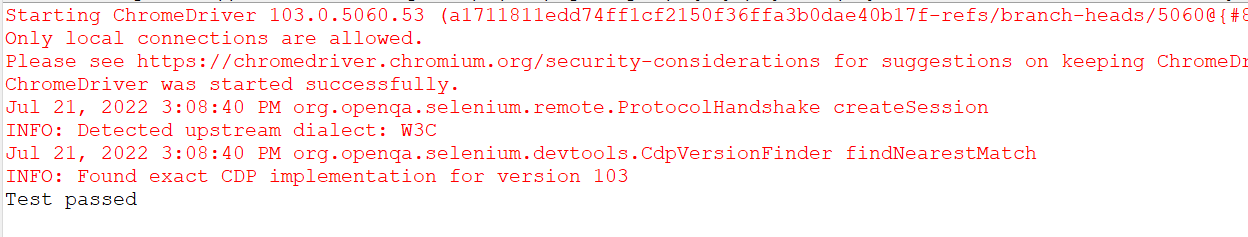
**Example:**

**Test Case 1**

**Code**



**Eg.Screenshot**



**Eg.Test Report**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case 1** | | | | | |
| **Project Name:** | | | | | |
| **Login Test Case** | | | | | |
| **Test Case ID: Test\_1** | | | **Test Designed By:** | | |
| **Test Priority(Low/Medium/High):** | | | **Test Designed Date:** | | |
| **Module Name**: | | | **Test Executed By :** | | |
| **Test Title :** | | | **Test Execution Date:** | | |
| **Description:** | | |  | | |
| **Pre-Condition :**User has valid username and password | | | | | |
| **Step** | **Test Step** | **Test Data** | **Expected Result** | **Actual Result** | **Status(Pass/**  **Fai l)** |
| 1 |  |  |  |  |  |
| 2 |  |  |  |  |  |
| 3 |  |  |
| 4 |  |  |
| 5 |  |  |  |  |  |
|  |  |  |  |  |  |
| 6 |  |  |
| 7 |  |  |  |  |  |
|  |  |  |  |  |
| **Post-Condition:** | | | | | |

**Test Case 2:**

**Code**

**Screenshot**

**Test report**

**Minimum 4 test cases (1 login 3 functionalities)**

# CHAPTER 6

# IMPLEMENTATION

## INTRODUCTION

Explanation

## IMPLEMENTATION PROCEDURES

Explanation

### User Training

Explanation

### Training on the Application Software

Explanation

### System Maintenance

Explanation

* + 1. **Hosting**

Explanation

**Eg.000Webhost**

Explanation

**Procedure for hosting a website on 000Webhost:**

Step 1: explanation

Step 2: explanation

Step 3: explanation

.

.

**Hosted Website:**

**Hosted Link: https://abc.000webhostapp.com**

**Hosted Link QR Code**

**Screenshot**

# CHAPTER 7

# CONCLUSION AND FUTURE SCOPE

## CONCLUSION

## 

.

* 1. **FUTURE SCOPE**

.

# CHAPTER 8

# BIBLIOGRAPHY

### REFERENCES:

* + - ..#Books
    - ..
    - ..
    - ..
    - ...

### WEBSITES:

* + - [..](http://www.w3schools.com/)
    - [..](http://www.jquery.com/)
    - [..](http://homepages.dcc.ufmg.br/%7Erodolfo/es-1-03/IEEE-Std-830-1998.pdf)
    - [..](http://www.agilemodeling.com/artifacts/useCaseDiagram.html)

# CHAPTER 9

# APPENDIX

## Sample Code

Main functionalities

## Screen Shots