DECLARATION

This is to declare that this project titled "**DESIGN AND IMPLEMENTATION OF BAKERY MANAGEMENT SYSTEM"** is written by **FOBASSO TOUTSOP BOREIL,** a student of the department of **SOFTWARE ENGINEERING at AZIMUT HIGHER INSTITUTE**. All statements made and conclusions drawn are outcomes of research works, and all borrowed ideas have been acknowledged by means of references quotations

AUTHOR:	**SUPERVISOR**
FOBASSO TOUOTSOP BOREIL	**MR. ATOH DENIS**
SIGNATURE: ……	**SINATURE……**
DATE……..	**DATE……**

CERTIFICATION

This is to certify that this work titled **"DESIGN AND IMPLEMENTATION OF A BAKERY MANAGEMENT SYSTEM"** has been written and all the research carried and presented by **FOBASSO TOUTSOP BOREIL** under the supervision of **Mr. ATOH DENIS** in achievement for the award of the **HIGHER NATIONAL DIPLOMA (HND).** It is approved for the contribution to scientific knowledge and literally presented.

ACADEMIC SUPERVISOR	**PROFESSIONAL SUPERVISOR**
Mr. ATOH DENIS	**Mr. TOUBE PIERRE**
SIGNATURE ……	**SIGNATURE…….**
DATE …….	**DATE …….**

DEDICATION

To my family

ACKNOWLEDGMENTS

- A work of such degree could not have been done and completed without any help and assistance. I address my sincere thanks to the almighty GOD for the strength and his grace in order to achieve this report.
- Special thanks go to my academic supervisor **Mr. TOUBE PIERRE** whose directives instructions encouragements and advice helped me to achieve this report.
- I would also like to thank all the personnel of **CIA CONSULTING** Enterprise for giving me the knowledge and for putting at my disposition the material necessary for me to achieve this report.
- I also thank particularly my parents and family for all their support both moral and material help, assistance, encouragements to all my brothers and sisters for their support and encouragements to all the persons who took of their time to answer my questionnaires.
- I would like to express my sincere gratitude to everyone who supported me during my internship.
- Furthermore, I would like to acknowledge the contributions of **Mr. ESSI JUNIOR** for his continuous support and guidance. Their expertise and willingness to share knowledge have been invaluable resources.
- I am truly grateful for the opportunity to be part of this internship program, and I look forward to applying the skills and knowledge gained in my future endeavors.

Thank you

FOBASSO TOUOTSOP BOREIL

ABSTRACT

In the world of business, bakeries rely heavily on management systems to streamline operations, track inventory, manage sales, and oversee customer orders efficiently. Many modern bakeries also integrate digital solutions for employee scheduling, financial tracking, and customer engagement. Planning for the necessary software, hardware, and digital tools must be an ongoing process to keep up with technological advancements in business management. By maintaining comprehensive records of transactions, inventory levels, and equipment usage, bakery owners can make informed decisions for improving efficiency and profitability. This report discusses the various types of records that should be kept in a bakery management system and how the information within these records can be applied to ongoing planning, decision-making, and operational improvements.

RESUME

Dans le monde des affaires, les boulangeries s'appuient fortement sur des systèmes de gestion pour rationaliser leurs opérations, suivre les stocks, gérer les ventes et superviser efficacement les commandes des clients. De nombreuses boulangeries modernes intègrent également des solutions numériques pour la planification du personnel, le suivi financier et l'engagement des clients. La planification des logiciels, du matériel et des outils numériques nécessaires doit être un processus continu afin de suivre les avancées technologiques dans la gestion des entreprises. En maintenant des registres complets des transactions, des niveaux de stock et de l'utilisation des équipements, les propriétaires de boulangeries peuvent prendre des décisions éclairées pour améliorer l'efficacité et la rentabilité. Ce rapport examine les différents types d'enregistrements à conserver dans un système de gestion de boulangerie et comment les informations contenues dans ces registres peuvent être appliquées à la planification, à la prise de décision et à l'amélioration des opérations.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

ABREVIATIONS	IMPLICATIONS
AJAX	Asynchronous JavaScript and XML
CDM	Conceptual data model
CSS	Cascading Style sheet
CV	Curriculum Vitae
DB	Database
DBMS	Database Management System
DC	Chart of classes
STD	State-Transition Diagram
GD	General Director
HR	Human resources
HND	Higher National Diploma
IFD	Information Flow Diagram
GLPI	Gestionaire libre de pack informatique
GUI	Graphical user interface
PC	Personal computer
OOP	Object oriented programming

RAD	Rapid Application Development
UML	Unified modeling language
OS	Operating system
API	Application programing interface
IOT	Internet of things

CHAPTER ONE:

GENERAL INTRODUCTION

INTRODUCTION

This report summarizes my internship experience at CIA CONSULTING, a software company which main focus was on implementation of web apps. I worked there as a web developer from July 8, 2024 to September 8, 2024. The purpose of this report is to describe the objectives, activities, challenges, achievements and recommendations of my internship. During the internship, we were assigned to the information and technology department, where they worked on various development projects, conducted quality assurance tests, and collaborated with cross-functional teams. The primary objectives of the internship included gaining hands-on experience in agile software development methodologies, enhancing coding skills, and understanding the software development life cycle and interacting with workers or personals on daily bases. This report will dive into the day-to-day activities, challenges faced, accomplishments achieved, and lessons learned during the internship period. By reflecting on the experiences at CIA CONSULTING, we aim to draw valuable insights and practical knowledge that will be beneficial for future career aspirations in the tech industry.

1.1 BACKGROUND OF THE STUDY

Software development is a structured process that involves creating, designing, testing, and maintaining software applications and systems. It has evolved significantly over the past few decades, reflecting advancements in both technology and methodologies. As organizations and individuals increasingly rely on technology, the demand for effective, efficient, and secure software continues to grow. Understanding the development process and the underlying principles is crucial for building reliable systems and applications.

1.1.1 Evolution of Software Development

The journey of software development dates to the mid-20th century, with the emergence of early computing systems. Initially, the software was manually written in machine language or assembly code. Early systems were often built for very specific tasks and lacked the complexity of modern applications. Over time, as computing technology advanced, the need for structured software development processes became more apparent. In the 1960s and 1970s, software engineering emerged as a discipline. The term "software engineering" was coined to describe the systematic approach needed to create large-scale software systems. One of the earliest methodologies to address this was the Waterfall Model, which follows a linear, step-by-step approach. However, as systems grew more complex and requirements changed more frequently, this rigid methodology proved inefficient. 1.1.2 Rise of Agile Methodologies In the 1990s, a shift toward more flexible and iterative approaches in software development took place. Agile methodologies, emphasizing adaptability, collaboration, and incremental progress, gained prominence. The Agile Manifesto, published in 2001, outlined key principles for software development, such as delivering working software frequently, responding to change, and valuing individuals and interactions over processes and tools. Scrum, Extreme Programming (XP), and Kanban are some of the widely adopted frameworks that focus on iterative cycles, short development sprints, and continuous improvement.

Agile has become the standard in many industries due to its ability to manage uncertainty and rapidly changing requirements, making it particularly suitable for fast-paced, dynamic environments. The focus on collaboration between developers, stakeholders, and customers ensures that the software developed aligns closely with user needs.

1.1.2 Key Phases in Software Development

Regardless of the methodology, software development generally involves several phases, including:

- **Requirements Gathering and Analysis:** Understanding the needs of the user or client is essential. This phase involves defining the functional and non-functional requirements, including performance, security, and scalability.

- **Design:** This phase includes creating architectural designs and selecting appropriate technologies to meet the specified requirements. The system design is often broken down into high-level design (system architecture) and low-level design (detailed component design).
- **Implementation:** The development team writes the code based on the design specifications. This phase may involve programming in various languages such as Java, Python, or C++ and using integrated development environments (IDEs) or version control systems (e.g., Git).
- **Testing:** Comprehensive testing ensures that the software is free from defects and functions as intended. Testing can include unit testing, integration testing, system testing, and acceptance testing. Automation tools and frameworks are often used to improve efficiency.
- **Deployment and Maintenance:** After testing, the software is deployed to the production environment, where it is used by the end users. Continuous monitoring, bug fixing, and performance improvements are part of ongoing maintenance.

1.1.3 Software Development Tools and Technologies

Modern software development is supported by a wide range of tools and technologies that enhance productivity, quality, and collaboration:

- **Version Control Systems (VCS):** Git and platforms like GitHub or GitLab allow developers to track code changes, collaborate with other developers, and maintain code integrity.
- **Integrated Development Environments (IDEs):** Tools like Visual Studio, IntelliJ IDEA, and Eclipse provide code editing, debugging, and testing functionalities in one place.
- **Continuous Integration/Continuous Deployment (CI/CD):** These practices automate the building, testing, and deployment processes, ensuring faster and more reliable software delivery.
- **Frameworks and Libraries:** Software frameworks (e.g., React, Laravel, Django, Spring) provide reusable code that accelerates development and ensures consistency.
- **Cloud Computing and DevOps**: Cloud platforms (e.g., AWS, Azure, Google Cloud) provide scalable infrastructure, while DevOps practices focus on automating and integrating the processes between software development and IT operations.

1.1.4 Challenges in Software Development

Software development, while essential, is not without its challenges. These include:

- **Complexity:** As software systems become more sophisticated, managing complexity becomes more difficult. Ensuring maintainability, scalability, and performance in large systems can be a major challenge.
- **Security:** Ensuring the security of software systems is a critical concern, especially in an era of rising cyber threats. Incorporating security measures into the development lifecycle (DevSecOps) is necessary to protect user data and maintain trust.
- **Changing Requirements:** Software projects often suffer from scope creep, where requirements change frequently or are poorly defined. Agile methodologies help manage this but still require strong communication and clear expectations.
- **Resource Management:** Managing resources, including developers, time, and budget, is vital for delivering a project on schedule. Delays, mismanagement, and underestimation can lead to project
  failure.

1.1.5 Impact of Software Development on Society

The impact of software development on society is profound. Software is the backbone of numerous industries such as healthcare, finance, education, and entertainment. It powers everything from personal devices like smartphones to critical systems used in transportation and communications. The digital transformation enabled by software development has led to significant economic growth, increased efficiency, and more personalized user experiences.

However, software development also raises ethical questions. Issues such as data privacy, algorithmic biases, and the environmental impact of software and hardware systems are important concerns. Developers are increasingly called to consider the societal and ethical implications of the technologies they create.

Conclusion

The field of software development is continuously evolving to meet the demands of an increasingly digital world. From its early days of simple programs to the complex, global-scale systems of today, software development plays a critical role in shaping modern society. While challenges exist, advancements in methodologies, tools, and technologies continue to drive

innovation and ensure that software can meet the needs of users and businesses alike. The future of software development is intertwined with emerging trends such as artificial intelligence, machine

learning, blockchain, and quantum computing, which will shape the next generation of applications and systems.

1.1 PROBLEMS STATEMENT

Here are some problems encountered in the management of a bakery in the traditional way

1. **Inventory Management Challenges**

The bakery faces challenges in tracking inventory, leading to overstocking or understocking of ingredients. This results in wastage, increased costs, and disruptions in production schedules.

2. **Order Management Issues**

Manual order handling often results in miscommunication, delayed order processing, and errors in fulfilling customer requirements, affecting customer satisfaction and retention.

3. **Inefficient Staff Scheduling**

The bakery struggles to manage staff schedules effectively, leading to overstaffing during low demand hours and understaffing during peak hours, causing inefficiencies and reduced productivity.

4. **Lack of Real-Time Sales Analytics**

The absence of a real-time sales tracking system prevents the bakery from identifying bestselling items, understanding customer preferences, and making data-driven decisions to optimize offerings.

5. **Poor Customer Relationship Management**

The bakery lacks a system to maintain customer data, track loyalty, and manage feedback, leading to missed opportunities for personalized marketing and improved customer engagement.

6. **Ineffective Waste Management**

The bakery does not have a system to monitor and minimize waste effectively, which contributes to higher operational costs and environmental concerns

7. **Difficulty in Menu Updates**

The bakery faces challenges in updating its menu for seasonal items, offers, or price changes due to the lack of a centralized, easily modified system.

8. **Compliance with Food Safety Standards**

Ensuring compliance with food safety regulations is difficult without a proper tracking and reporting mechanism, which could expose the bakery to regulatory risks.

1.2 OBJECTIVES OF THE STUDY

1.2.1 Main research objectives

The design and implementation of a bakery management system to ease order placing by customers and facilitate inventory management

1.2.2 Specific research objectives

During this internship period our main research objectives where to;

Objective 1: the design and implementation of a user-friendly interface for inventory management employees tracking and real-time analytics.

Objective 2: To implement a bakery management system that streamlines core operations and enhances overall efficiency.

Objective 3: To improve inventory management and reduce wastage through an automated tracking system.

Objective 4: To enhance customer satisfaction by simplifying order management and integrating online ordering options.

Objective 5: Develop reporting tools that generate detailed analytics, performance metrics, and historical data on sails informed decision-making, resource planning, and identification of trends for process optimization.

Objective 6: To optimize staff scheduling and resource allocation for improved productivity.

Objective 7: To ensure the scalability and adaptability of the system for future business growth and changing requirements.

Objective 8: Implement real-time updates and notification systems within the software to alert relevant stakeholders about task assignments, completions, delays, or critical issues, ensuring timely communication and decision-making.

Objective 9: Conduct extensive user testing sessions with maintenance personnel, supervisors, and administrators to gather feedback, identify usability issues, and iteratively improve the software based on user perspectives and suggestions.

1.3 Research questions

1. How can a digital bakery management system improve operational efficiency and customer satisfaction in a small-to-medium-sized bakery?

2. What impact does an automated inventory management system have on reducing wastage and ensuring the availability of essential ingredients in a bakery?

3. How can integrating an online ordering platform with a bakery management system enhance customer convenience and loyalty?

4. How can real-time sales analytics help a bakery identify trends and optimize its product offerings to maximize profitability?

5. What role can a digital management system play in minimizing waste and promoting sustainability in a bakery's operations?

6. How can an automated staff scheduling feature improve labor utilization and reduce inefficiencies in a bakery?

7. Can a bakery management system help streamline compliance with food safety standards and reduce the risk of violations?

8. How does the integration of online ordering and in-store operations in a bakery management system impact overall business performance?

9. How does the implementation of a digital payment system in a bakery management system influence transaction speed and customer satisfaction?

10. What design features of a bakery management system can ensure scalability and adaptability to meet the evolving needs of a growing bakery business?

1.4 SIGNIFICANCE OF STUDY

1. Operational Efficiency: The system streamlines day-to-day operations, such as inventory management, order processing, and staff scheduling, reducing manual errors and saving time.

2. Cost Savings: Automated inventory tracking helps minimize wastage and optimize resource utilization, leading to lower operational costs.

3. Data-Driven Decisions: Real-time sales analytics enable the bakery to identify trends, forecast demand, and optimize product offerings for higher profitability.

4. Improved Customer Experience: Features like online ordering, faster payments, and personalized loyalty programs enhance customer satisfaction and retention.

5. Simplified Workflows: Automation of repetitive tasks reduces workload, allowing staff to focus on customer service and product quality.

6. Optimized Scheduling: A staff scheduling feature ensures balanced workloads, leading to higher productivity and job satisfaction.

7. Convenience: Online ordering and faster checkout processes improve the overall customer experience.

8. Personalization: Enhanced customer relationship management allows for personalized offers and better customer engagement.

9. Foundation for Innovation: This study serves as a reference for developing or improving management systems in other businesses, particularly in the food and beverage industry.

10. Scalability Insights: Insights into scalability and adaptability can guide similar projects for larger businesses or franchises.

11. Adoption of Digital Tools: The study highlights the potential of digital systems in improving small and medium-sized enterprises (SMEs), encouraging more businesses to adopt technology for growth.

1.5 SCOPE AND LIMITATIONS

1.5.1 SCOPE

The Bakery Management System is designed to streamline bakery operations by automating key processes such as inventory management, sales tracking, order processing, and customer management. The system will provide an intuitive interface for bakery staff to efficiently manage daily tasks while improving accuracy and reducing manual errors.

1. Inventory Management: Tracks raw materials and finished products, updating stock levels in real-time to prevent shortages or wastage.

2. Order Processing: Manages customer orders, both in-store and online, ensuring timely fulfillment and accurate record-keeping.

3. Sales Tracking & Reporting: Generates sales reports and analytics, allowing bakery owners to monitor revenue, best-selling products, and financial trends.

4. User Management: Supports different access levels for admins, bakers, and cashiers, ensuring secure and efficient system use.

5. Expense & Cost Management: Tracks production costs, ingredient usage, and operational expenses to help optimize pricing strategies.

6. Customer Management: Maintains customer records, order history, and preferences to enhance customer service and marketing efforts.

1.5.2 SCOPE

It is important to consider the limitations or challenges that may arise during the development and implementation process. Here are a few potential limitations that we considered:

1. Limited Scope: The system is designed specifically for bakery operations, which may limit its applicability to other types of businesses without significant modifications.

2. Resource Constraints: The development and implementation process may face challenges due to limited financial resources, hardware availability, or technical expertise.

3. Time Constraints: The time allocated for the development and testing phases of the project may not be sufficient to explore all potential features or address all possible use cases.

4. User Adoption Challenges: Employees or stakeholders unfamiliar with digital tools may face a learning curve, which could delay full adoption of the system.

5. Data Dependence: The system's effectiveness depends on accurate and timely input of data, which may not always be ensured due to human error or neglect.

6. Scalability Issues: While designed with scalability in mind, the system may require further enhancements to support a significantly larger customer base or additional bakery branches.

7. Internet Dependency: Features like online ordering and cloud-based data storage rely on stable internet connections, which may not be consistently available in all locations.

8. Security Concerns: If adequate security measures are not implemented, sensitive customer and business data might be vulnerable to breaches or misuse.

9. Regulatory Compliance: The system may not fully address all region-specific regulatory requirements, such as food safety standards or data privacy laws, without additional customization.

10. Data security and privacy: Recording sails activities involves handling sensitive data. Ensuring data security, privacy, and compliance with relevant regulations (e.g., GDPR) can be a significant challenge that needs to be addressed.

1.6 OPERATIONAL DEFINITIONS OF TERMS

1. Bakery Management System: A software application designed to automate and streamline bakery operations, including inventory tracking, order processing, sales management, and customer handling.

2. Inventory Management: The process of monitoring and controlling raw materials and finished goods to ensure efficient stock usage and minimize waste.

3. Order Processing: The workflow of receiving, managing, and fulfilling customer orders, whether in-store or online.

4. Sales Tracking: The process of recording and analyzing sales transactions to monitor business performance and identify trends.

5. Laravel: A PHP framework used for developing web applications, providing built-in tools for routing, authentication, database management, and more.

6. Bootstrap: A front-end framework that helps design responsive and visually appealing web interfaces using pre-built CSS and JavaScript components.

7. User Management: A system feature that assigns different access levels to users (e.g., admin, customers) to ensure security and role-based functionality.

8. Customer Management: The system's ability to store customer details, order history, and preferences to enhance personalized service and marketing.

CHAPTER 2 :

LITTERATURE REVIEW

2.1 THEORITICAL REVIEW

Computer science, as a rapidly evolving field at the forefront of technological innovation, plays a pivotal role in shaping the digital landscape and driving advancements in various domains. This literature review aims to explore the diverse spectrum of research, theories, methodologies, and applications within computer science, spanning from foundational principles in algorithms and data structures to cutting-edge developments in artificial intelligence, cyber security, and software engineering. By synthesizing existing literature, analyzing key trends, and identifying research gaps, this review seeks to offer a comprehensive overview of the current state of computer science research, highlighting emerging paradigms, challenges, and opportunities for future exploration. In this chapter we are going to talk about some theoretical concept related to our work. We are also going to analyses some existing solutions to bring out their strength and weakness.

In this section we are going to discuss some concepts and technologies we will use in our work and their different advantages and disadvantages.

2.1.1 VISUAL STUDIO CODE

Visual Studio is an Integrated Development Environment (IDE) developed by Microsoft to develop Desktop applications, GUI (Graphical User Interface), console, web applications, mobile applications, cloud, and web services, etc. With the help of this IDE, you can create managed code as well as native

code. It uses the various platforms of Microsoft software development software like Windows store, Microsoft Silverlight, and Windows API. It is not a language-specific IDE as you can use this to write code in C#, C++, VB (Visual Basic), Python, JavaScript, and many more languages. It provides support for 36 different programming languages. It is available for Windows as well as for macOS. (https://www.geeksforgeeks.org/introduction-to-visual-studio/, 2024)

Evolution of Visual Studio

The first version of VS (Visual Studio) was released in 1997, named as Visual Studio 97 having version number 5.0. The latest version of Visual Studio is 15.0 which was released on March 7, 2017. It is also termed as Visual Studio 2017. The supported .Net Framework Versions in latest Visual Studio is 3.5 to 4.7. Java was supported in old versions of Visual Studio but in the latest version doesn't provide any support for Java language.

Visual Studio Editions

There are 3 editions of Microsoft Visual Studio as follows**:**

1. Community

It is a free version which is announced in 2014. All other editions are paid. This contains the features similar to Professional edition. Using this edition, any individual developer can develop their own free or paid apps like .Net applications, Web applications and many more. In an enterprise organization, this edition has some limitations. For example, if your organization have more than 250 PCs and having annual revenue greater than $1 Million (US Dollars) then you are not permitted to use this edition. In a non-enterprise organization, up to five users can use this edition. Its main purpose is to provide the Ecosystem (Access to thousands of extensions) and Languages (You can code in C#, VB, F#, C++, HTML, JavaScript, Python, etc.) support.

2. Professional

It is the commercial edition of Visual Studio. It comes in Visual Studio 2010 and later versions. It provides the support for XML and XSLT editing and includes the tool like Server Explorer and integration with Microsoft SQL Server. Microsoft provides a free trial of this edition and after the trial period, the user has to pay to continue using it. Its main purpose is to provide Flexibility (Professional

developer tools for building any application type), Productivity (Powerful features such as Code Lens improve your team's productivity), Collaboration (Agile project planning tools, charts, etc.) and Subscriber benefits like Microsoft software, plus Azure, Plural sight, etc.

3. Enterprise

It is an integrated, end-to-end solution for teams of any size with the demanding quality and scale needs. Microsoft provides a 90-days free trial of this edition and after the trial period, the user has to pay to continue using it. The main benefit of this edition is that it is highly scalable and deliver high-quality software.

Getting Started with Visual Studio 2017

• First, you have to download and install the Visual Studio. For that, you can refer to Downloading and Installing Visual Studio 2017. Don't forget to select the .NET core workload during the installation of VS 2017. If you forget then you have to modify the installation.

• You can see a number of tool windows when you will open the Visual Studio and start writing your first program as follows:

Code Editor: Where the user will write code.

• Output Window: Here the Visual Studio shows the outputs, compiler warnings, error messages and debugging information.

• Solution Explorer: It shows the files on which the user is currently working.

• Properties: It will give additional information and context about the selected parts of the current project.

• A user can also add windows as per requirement by choosing them from View menu. In Visual Studio the tool windows are customizable as a user can add more windows, remove the existing open one or can move windows around to best suit.

• Various Menus in Visual Studio: A user can find a lot of menus on the top screen of Visual Studio as shown below

• Create, Open and save projects commands are contained by File menu.

• Searching, Modifying, Refactoring code commands are contained by the Edit menu.

• View Menu is used to open the additional tool windows in Visual Studio.

• Project menu is used to add some files and dependencies in the project**.**

• To change the settings, add functionality to Visual Studio via extensions, and access various Visual Studio tools can be used by using Tools menu.

• The below menu is known as the toolbar which provide the quick access to the most frequently used commands. You can add and remove the commands by going to View → Customize

ADVANTAGES OF USING VISUAL STUDIO IDE

- A full-featured programming platform for several operating systems, the web, and the cloud, Visual Studio IDE is available. Users can easily browse the UI so they can write their code quickly and precisely.
- To help developers quickly identify potential errors in the code, Visual Studio offers a robust debugging tool.
- Developers can host their application on the server with confidence because they have eliminated anything that could lead to performance issues.
- No matter what programming language developers are using, users of Visual Studio can get live coding support. For faster development, the Platform offers an autocomplete option. The built-in intelligent system offers descriptions and tips for APIs.
- Through Visual Studio IDE you can easily collab with your teammates in a same project. This IDE helps the developers to share, push and pull their code with their teammates.
- Every user of Visual Studio has the ability to customize it. They have the option to add features based on their needs. For example, they can download add-ons and install extensions in their IDE. Even programmers can submit their own extensions.

DISADVANTAGES OF USING VISUAL STUDIO IDE

- **Resource Consumption:** Since VS Code is built on top of Chromium, it can consume more memory and CPU resources compared to lightweight text editors. Some users may find it slower or resource-intensive.
- **Telemetry:** By default, VS Code collects telemetry data to improve the product. While this helps the development team, some users may be concerned about privacy.
- **Lack of Integrated Help:** Unlike some IDEs, VS Code lacks an integrated help system that directly links to tutorials, videos, or documentation. Users need to rely on external resources.
- **Limited Built-in Features:** While extensions enhance functionality, some users might miss certain features commonly found in full IDEs, such as built-in database tools or advanced project management.
- **Initial Learning Curve:** The extensive range of features can be overwhelming for beginners. However, once users become familiar with the tool, it becomes more intuitive.
- **No Native Cloud Settings Sync:** Although there are workarounds, VS Code does not natively provide cloud-based

Note:

- Support for different programming languages in Visual Studio is added by using a special Package which is known as Language Service.
- When you will install the Visual Studio then the functionality which is coded as VSPackage will be available as Service.
- Visual Studio IDE provides the three different types of services known as SVsSolution, SVsUIShell, and SVsShell.
- SVsSolution service is used to provide the functionality to enumerate solutions and projects in Visual Studio.
- SVsUIShell service is used to provide User Interface functionality like toolbars, tabs etc.
- SvsShell service is used to deal with the registration of VSPackages.

2.1.2 WEB APPLICATION DEVELOPMENT

Web application development is the creation of application programs that reside on remote servers and are delivered to the user's device over the Internet. A web application (web app) does not need to be downloaded and is instead accessed through a network. An end user can access a web

Application through a web browser such as Google Chrome, Brave, Safari, or Mozilla Firefox. A majority of web applications can be written in JavaScript, Cascading Style Sheets (CSS), and HTML5. Web application development will typically have a short development life-cycle lead by a small development team. Front-end development for web applications is accomplished through client-side programming. Clients refer to a computer application such as a web browser. Client-side programming will typically utilize HTML, CSS and JavaScript. HTML programming will instruct a browser how to display the on-screen content of web pages, while CSS keeps displayed information in the correct format. JavaScript will run JavaScript code on a web page, making some of the content interactive. Server-side programming powers client-side programming and is used to create the scripts that web applications use. Scripts can be written in multiple scripting languages such as Ruby, Java and Python. Server-side scripting will create a custom interface for the end-user and will hide the source code that makes up the interface (Web application development, 2023).A database such as MySQL, SQLITE3 or Mongo DB can be used to store data in web application development. There are many tools and programming languages that can be used to create dynamic and interactive web pages. PYTHON, PHP, ASP.NET and JSP are the most popular programming tools for develop web pages.

a. PHP

PHP (Hypertext Preprocessor) is a popular open-source server-side scripting language designed specifically for web development. It is embedded within HTML, allowing developers to create dynamic and interactive web pages efficiently. PHP excels in handling server-side tasks like managing databases, processing forms, and generating dynamic content.

Some of the companies that use PHP as their technology stacks include**:**

1. Facebook: One of the largest social media platforms in the world. Facebook initially used PHP extensively to build its dynamic web interface. Although it now uses a custom version called Hack, PHP remains the foundation of its development.

2. WordPress: The most popular Content Management System (CMS), powering over 40% of websites globally. PHP forms the core of WordPress, enabling the creation and management of blogs, e-commerce sites, and corporate websites with themes and plugins.

3. Wikipedia: The world's largest online encyclopedia. Wikipedia uses PHP to manage its extensive content, ensure fast page loads, and handle its massive global traffic.

4. Slack: A leading team collaboration and messaging platform. PHP is used in Slack's back-end infrastructure for certain operations,

A. ADVANTAGES OF PHP

Despite the fact that PHP has been around for a long time and is well known to everyone, there is often a gap in understanding its advantages over other programming languages, especially when used in web development. Let’s list some of these advantages.

1. Open Source and Free: PHP is open-source and available for free, making it cost-effective for businesses and developers. The large community contributes to constant updates, resources, and support.

2. Platform Independence: PHP runs on all major operating systems, including Windows, Linux, and macOS. Its cross-platform compatibility ensures seamless integration with various systems and web servers like Apache and Nginx.

3. Ease of Learning and Use: PHP has a simple and intuitive syntax, making it beginner-friendly. Developers with basic programming knowledge can quickly learn and start building applications with PHP.

4. Rich Ecosystem of Frameworks: PHP offers a variety of frameworks like Laravel, Symfony, CodeIgniter, and Yii. These frameworks simplify development by providing built-in features for authentication, routing, and database management.

5. Extensive Database Support: PHP supports a wide range of databases, including MySQL, PostgreSQL, SQLite, Oracle, and MongoDB. This versatility allows developers to choose the best database for their application.

6. Large Community and Resources: PHP boasts a vast community of developers, resulting in abundant resources, tutorials, forums, and libraries. This makes troubleshooting and learning more accessible.

7. Scalability: PHP supports building scalable web applications, from small blogs to enterprise-level systems like Facebook and Wikipedia. Frameworks and tools further enhance its scalability.

8. Fast Development and Deployment: PHP enables rapid application development with its built-in functions, extensive libraries, and frameworks. It is especially efficient for small to medium-scale applications.

9. Integration with Other Technologies: PHP integrates seamlessly with various technologies, including HTML, CSS, JavaScript, and XML. It also works well with APIs, web services, and CMS platforms.

10. High Performance: With modern versions (PHP 7 and PHP 8), PHP has significantly improved in performance. Features like Just-In-Time (JIT) compilation enhance execution speed, making it suitable for high-traffic applications.

11. Security Features: PHP includes built-in security features, such as input validation, encryption, and mechanisms to prevent SQL injection, XSS, and CSRF attacks. Frameworks further enhance security with pre-configured tools.

12. Widespread Hosting Support: Most web hosting providers offer PHP support out of the box, making it easier to deploy applications without special configurations.

11. Web Development Using PHP Frameworks

Another good thing about web development using PHP is that it has many frameworks that simplify the development process which provide pre-built tools, libraries, and structures to streamline the development process, enhance productivity, and promote best practices like MVC (Model-

ViewController) architecture. Depending on what you're doing, you may need different frameworks. Let's take a look at the most well-known PHP frameworks

LARAVEL

Laravel is a popular open-source PHP framework designed for web application development. It follows the Model-View-Controller (MVC) architecture, which helps in organizing code efficiently and maintaining a clear separation between logic and presentation. Laravel is widely used for building scalable and secure applications due to its elegant syntax and developer-friendly features.

One of the key strengths of Laravel is its Eloquent ORM, which simplifies database interactions by allowing developers to work with database records using an intuitive syntax. The framework also includes the Blade templating engine, which helps in creating dynamic and reusable views with minimal overhead. Additionally, Laravel's routing system enables easy management of web routes, making it easier to define application functionalities. Security is another major advantage, as Laravel provides built-in authentication, hashing, and protection against CSRF and SQL injections. The Artisan CLI tool further enhances productivity by automating repetitive tasks such as database migrations and seeding. Furthermore, Laravel includes a queue system that manages background jobs, improving application performance.

For a bakery management system, Laravel offers efficiency, scalability, and security, making it an ideal choice. Its built-in tools streamline the development process, allowing for faster implementation of features. The framework ensures scalability, making it suitable for bakeries of all sizes, from small shops to large franchises. Laravel also provides strong security measures to protect sensitive data, such as customer orders and payment details. Additionally, it is API-ready, allowing integration with mobile applications and third-party services like payment gateways and inventory systems.

When implemented in a bakery management system, Laravel can facilitate user authentication, ensuring secure access for administrators, employees, and customers. It also supports product and inventory management, allowing bakery owners to add, update, and track stock efficiently. The framework enables seamless order processing, handling customer purchases, invoices, and receipts effortlessly. Additionally, Laravel can generate sales reports

and analytics, helping bakery owners make data-driven decisions to optimize business performance. With its extensive features and robust architecture, Laravel is an excellent choice for developing a bakery management system.

B. DISADVANTAGES

1. Inconsistent Syntax: PHP's syntax has evolved over the years, leading to inconsistencies that can confuse developers. Leading errors and increased time spent learning or debugging code.

2. Security Vulnerabilities: While PHP provides security features, its ease of use and prevalence make it a common target for attackers. Applications may become vulnerable to SQL injection, XSS, and other attacks if developers do not follow best practices.

3. Performance Limitations: PHP is interpreted at runtime, which may lead to slower performance compared to compiled languages like Java or C++. While modern versions (PHP 7 and PHP 8) have improved performance, it might not be the best choice for extremely high-performance applications.

4. Lack of Modular Design: Older PHP projects often suffer from a lack of modularity and separation of concerns. Maintaining and scaling such applications can be difficult without frameworks or proper coding practices.

5. Limited Suitability for Modern Needs: PHP was originally designed for server-side scripting and may not cater as effectively to modern, real-time, or microservices architectures. Developers may prefer other languages like Node.js for building modern, real-time applications.

6. Weak Typing: PHP's dynamic typing can lead to unexpected behaviors and subtle bugs, especially in large, complex applications. Debugging can become cumbersome, and enforcing type safety may require additional effort.

7. Over-Reliance on Community Contributions: PHP's extensive ecosystem depends heavily on third-party libraries and frameworks. Poorly maintained libraries or lack of official support can lead to compatibility and security issues.

8. Perception as Outdated: PHP is often viewed as an older technology compared to newer languages like Python, Ruby, or JavaScript. This perception might make it harder to attract developers to PHP projects.

9. Poor Error Handling in Older Versions: Error handling in PHP (prior to modern versions) was not as robust or developer friendly. Debugging and managing errors were more difficult, though this has improved with newer versions.

10. Tight Coupling with Web Development: PHP is primarily designed for web development, making it less suitable for non-web applications like standalone software or mobile apps. Developers seeking to create versatile or hybrid applications may choose other languages.

B. HTML 5

- **HTML5 (Hypertext Markup Language 5)** is a markup language used for structuring and presenting hypertext documents on the World Wide Web. It was the fifth and final[4] major HTML version that is now a retired World Wide Web Consortium (W3C) recommendation. The current specification is known as the HTML Living Standard. It is maintained by the Web Hypertext Application Technology Working Group (WHATWG), a consortium of the major browser vendors (Apple, Google, Mozilla, and Microsoft).
- **HTML5** was first released in a public-facing form on 22 January 2008,[2] with a major update and "W3C Recommendation" status in October 2014.[5][6] Its goals were to improve the language with support for the latest multimedia and other new features; to keep the language both easily readable by humans and consistently understood by computers and devices such as web browsers, parsers, etc., without XHTML's rigidity; and to remain backward-compatible with older software. HTML5 is intended to subsume not only HTML 4 but also XHTML1 and even the DOM Level 2 HTML itself.
- **HTML5** includes detailed processing models to encourage more interoperable implementations; it extends, improves, and rationalizes the markup available for documents and introduces markup and application programming interfaces (APIs) for complex web applications. For the same reasons, HTML5 is also a candidate for cross-platform mobile applications because it includes features designed with low-powered devices in mind.

C. DATA MANAGEMENT SYSTEM

Database Management Systems (DBMS) are software systems used to store, retrieve, and run queries on data. A DBMS serves as an interface between an end-user and a database, allowing users to create, read, update, and delete data in the database. They manage the data, the database engine, and the database schema, allowing for data to be manipulated or extracted by users and other programs. This helps provide data security, data integrity, concurrency, and uniform data administration procedures.

DBMS optimizes the organization of data by following a database schema design technique called normalization, which splits a large table into smaller tables when any of its attributes have redundancy in values. DBMS offer many benefits over traditional file systems, including flexibility and a more complex backup system (What is Database Management Systems (DBMS)? 2023). Database management systems can be classified based on a variety of criteria such as the data model, the database distribution, or user numbers. The most widely used types of DBMS software are relational, distributed, hierarchical, object-oriented, and network.

A distributed DBMS is a set of logically interrelated databases distributed over a network that is managed by a centralized database application. This type of DBMS synchronizes data periodically and ensures that any change to data is universally updated in the database (What is Database Management Systems (DBMS)?, 2023**).**

Hierarchical databases organize model data in a tree-like structure. Data storage is either a top-down or bottom-up format and is represented using a parent-child relationship. The network database model addresses the need for more complex relationships by allowing each child to have multiple parents. Entities are organized in a graph that can be accessed through several paths. Relational database management systems (RDBMS) are the most popular data model because of its user-friendly interface. It is based on normalizing data in the rows and columns of the tables. This is a viable option when you need a data storage system that is scalable, flexible, and able to manage lots of information. Object-oriented models store data in objects instead of rows and columns. It is based on object-oriented programming (OOP) that allows objects to have members such as fields, properties, and methods**.**

There is a wide range of database software solutions, including both enterprise and open-source solutions, available for database management. Here are some of the most popular database management systems:

Oracle Database is a commercial relational database management system. It utilizes enterprise-scale database technology with a robust set of features right out of the box. It can be stored in the cloud or on-premises.

MySQL is a relational database management system that is commonly used with open-source content management systems and large platforms like Facebook, Twitter, and YouTube. Developed by Microsoft, SQL Server is a relational database management system built on top of structured query language (SQL), a standardized programming language that allows database administrators to manage databases and query data**.**

SQLite3 is a database engine written in the C programming language. It is not a standalone app; rather, it is a library that software developers embed in their apps. As such, it belongs to the family of embedded databases. It is the most widely deployed database engine, as it is used by several of the top web browsers, operating systems, mobile phones, and other embedded systems. Many programming languages have bindings to the SQLite library. It generally follows Postgre SQL syntax, but does not enforce type checking by default. This means throne can, for example, insert a string into a column defined as an integer.

D. Mobile platforms

Android

Android is a mobile operating system that has been around for nearly 15 years. You'll primarily find it the base operating system of phones and tablets worldwide. Additionally, other operating systems natively support Android applications, including Chrome OS and Windows 11. Search giant Google owns this mobile OS (Mobile platforms, frameworks & environments, 2023). However, the system is open source, making it freely accessible to anyone, even for commercial use. This makes it very different from Apple's IOS, MACOS, and Microsoft's Windows, all closed-source platforms.

IOS

Android is by far the world's most popular operating system. Global Stats' Stat counter puts Android in the lead in mobile OS market share with 69.74% as of January 2022. This dwarfs Apple's iOS by a significant margin, which is the second-most-popular mobile operating system globally at 29.49% (Mobile platforms, frameworks & environments, 2023). There are over 2.6million applications available from the official Google Play Store, but you can also sideload apps from the web. This variety makes these phones very powerful and customizable — but also susceptible to viruses and other types of malwares. IOS, an operating system from Apple, was originally developed for the iPhone. Later it was extended to support iPod Touch, iPad and Apple TV. Apple's App Store contains more than 500,000 applications and boasts more than 25 billion downloads collectively. It holds the reputation of intelligent UI creator which is based on the concept of direct manipulation, using multi-touch gestures.

2.2 CONCEPTUAL REVIEW

Here we are going to show how the different components explained above can interact with each other and produce an overview of our project**.**

In a typical web application architecture, the system comprises several layers that work together to deliver a seamless user experience. The following is a textual representation of a web application architecture:

- **User Interface (UI):**

The front-end layer where users interact with the application. Built with HTML, CSS, and JavaScript. Responsive design for optimal user experience across devices.

- **Presentation Layer:**

Handles the logic related to the presentation and user interface.

Includes front-end frameworks like React, Angular, or Vue.js.

Responsible for rendering data received from the server.

- **Application Logic Layer:**

Manages the core functionality of the application.

Utilizes back-end frameworks like Node.js, Django, or Ruby on Rails.

Processes requests from the presentation layer and communicates with the database**.**

- **Web Server:**

Acts as an intermediary between the front-end and back-end.

Responds to HTTP requests from the client and routes them to the appropriate components**.**

- **API (Application Programming Interface):**

Defines how different software components should interact.

RESTful or GraphQL APIs are common for communication between the front-end and back-end.

Business Logic Layer:

- Implements the business rules and logic of the application.
- Validates and processes data before storing it in the database.

Database:

- Stores and manages the application's data.
- Relational databases like MySQL or PostgreSQL, or NoSQL databases like MongoDB are commonly used.

Data Storage:

- Includes file storage systems or cloud storage for storing user uploads, media files, etc.
- Authentication and Authorization:
- Manages user identity and access control.
- Ensures secure user authentication and authorization to access specific resources.

Security Layer:

- Implement security measures to protect against common web application vulnerabilities.
- Includes encryption, input validation, and other security protocols.

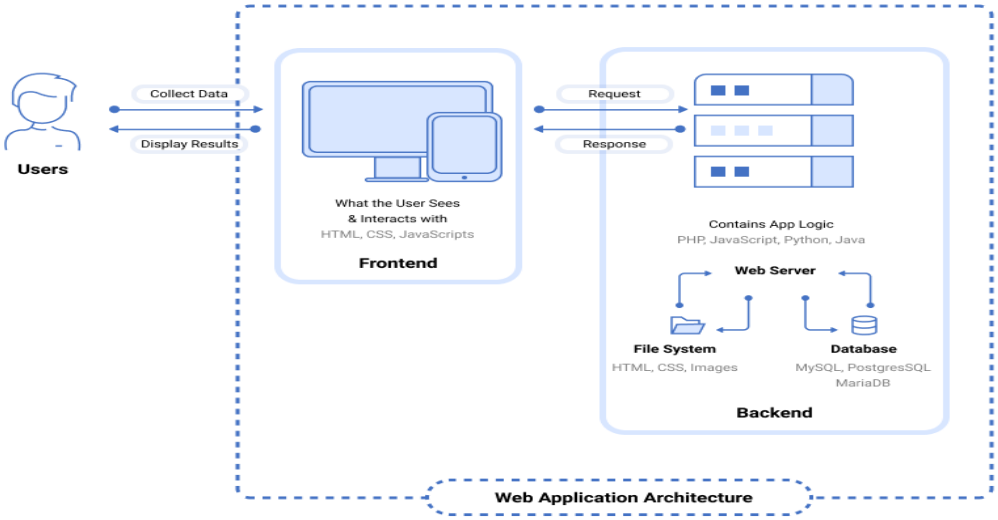


Figure 1 :3-tier architecture

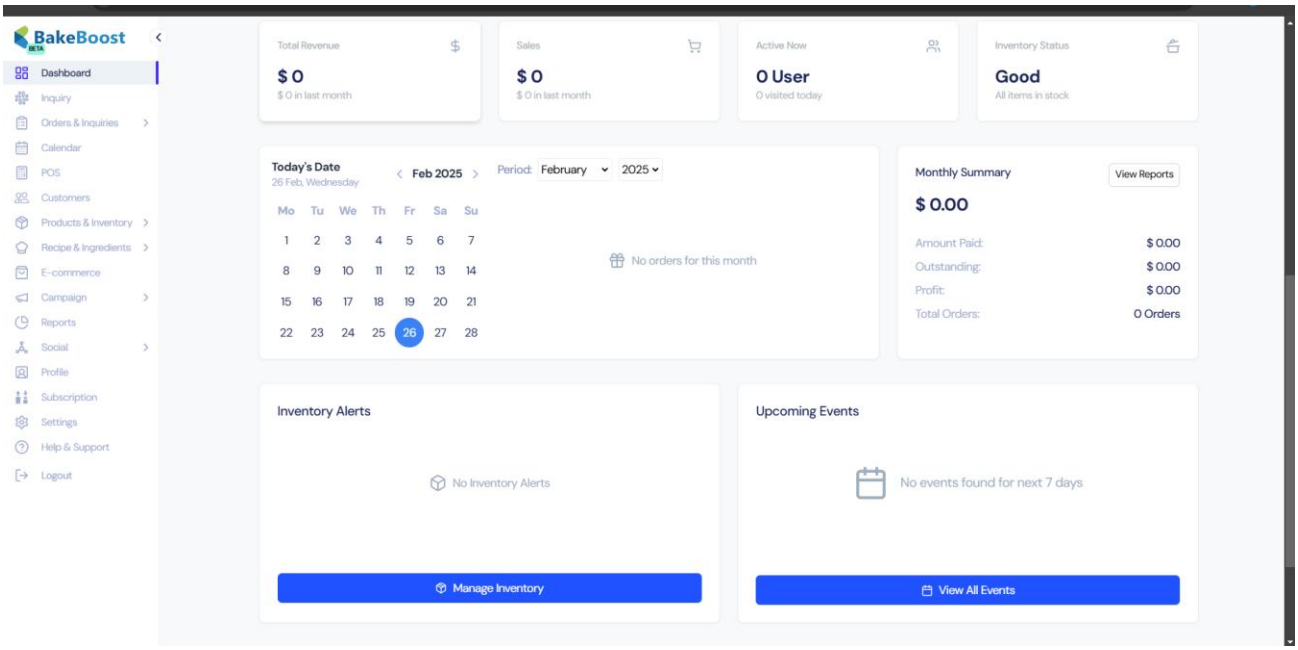
source(Altvater, 2017)

2.3 EMPERICAL REVIEW

In the world there are many companies that use softwares to manage the activities of their bakeries. In this section we are going to discuss existing softwares which help manage the activities of a bakery enterprise.

2.3.2 BakeBoost

It’s a comprehensive, cloud-based bakery management application designed to assist bakers—from home-based entrepreneurs to established bakery owners—in efficiently managing their operations. The platform offers a suite of integrated tools that streamline various aspects of bakery management, enhancing productivity and profitability.



Key Features:

- **Order Management**: Simplifies the process of receiving, tracking, and fulfilling customer orders from multiple channels, ensuring timely deliveries and accurate record-keeping.
- **Production Scheduling**: Utilizes an intuitive drag-and-drop calendar interface, allowing bakers to plan and schedule production tasks efficiently, manage ingredient requirements, and maintain compliance with industry standards.
- **Inventory Control**: Monitors stock levels of ingredients and finished products in real-time, helping to reduce waste and prevent shortages.
- **Customer Relationship Management (CRM)**: Maintains detailed customer profiles, including order histories and preferences, enabling personalized service and targeted marketing efforts.

- **Analytics and Reporting**: Generates comprehensive reports on sales, production, and inventory, providing valuable insights to inform business decisions.
- **Recipe Creation Module**: Allows bakers to create, store, and manage recipes in one centralized location, ensuring consistency and streamlining the production process.
- **Cost Management**: Assists in tracking expenses and optimizing resource allocation, effectively reducing costs and increasing revenues.
- **Delivery Management**: Facilitates the organization and tracking of deliveries, ensuring timely and accurate order fulfillment.

Advantages of BakeBoost

1. **Comprehensive Bakery Management** – BakeBoost provides an all-in-one platform for managing inventory, sales, orders, and production scheduling, reducing the need for multiple tools.
2. **Cloud-Based Accessibility** – Since BakeBoost is cloud-based, it allows bakery owners and staff to access the system from anywhere using any device with an internet connection.
3. **Automated Inventory Tracking** – The system helps monitor ingredient stock levels in real time, reducing waste and ensuring ingredients are always available when needed.
4. **Recipe and Cost Management** – BakeBoost allows bakers to create, store, and adjust recipes while automatically calculating production costs, helping to improve pricing strategies.
5. **Detailed Sales and Performance Reports** – The platform generates analytics and reports on sales, customer orders, and expenses, helping businesses make data-driven decisions.
6. **Customer Relationship Management (CRM)** – Stores customer order histories and preferences, making it easier to provide personalized services and marketing campaigns.
7. **User-Friendly Interface** – The intuitive dashboard and navigation make it easy for bakery owners and employees to manage operations without requiring extensive technical knowledge.

8. **Delivery and Order Management** – Enables businesses to track deliveries and orders efficiently, ensuring timely fulfillment and better customer satisfaction.

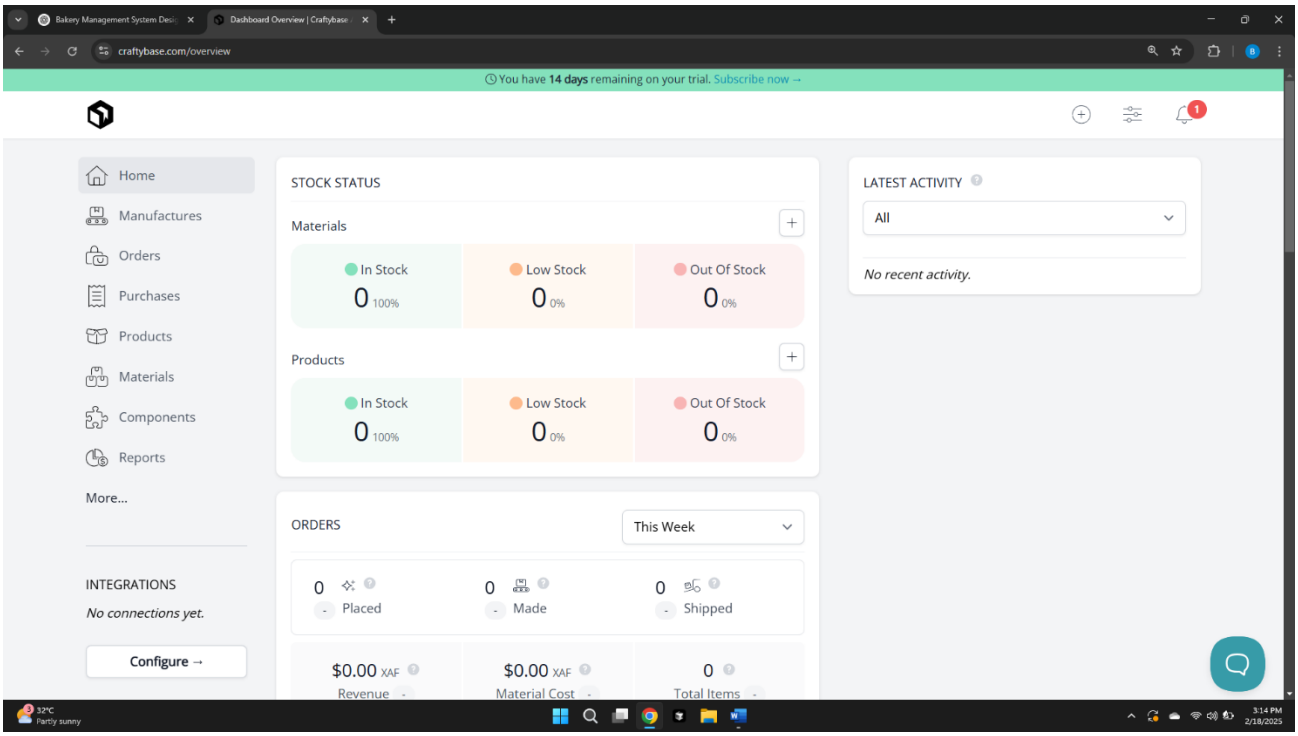
Disadvantages of BakeBoost

1. **Requires Internet Connection** – Since it is a cloud-based system, BakeBoost relies on a stable internet connection, making it less effective in areas with poor connectivity.
2. **Limited Free Version** – While BakeBoost offers a free plan, advanced features such as detailed analytics and multi-user access may require a paid subscription.
3. **Learning Curve for New Users** – Some bakery owners and staff who are not familiar with digital management tools may require time to adapt and learn the system.
4. **Customization Limitations** – While BakeBoost offers essential bakery management features, some businesses with unique requirements may find the customization options limited.
5. **Dependency on Third-Party Support** – If there are technical issues or system downtimes, businesses have to rely on BakeBoost's customer support, which might not always provide immediate assistance.

2.3.2 CraftyBase

Craftybase is a cloud-based inventory and production management system designed for small-scale manufacturers, including bakeries and handmade product businesses. This software is built to help businesses efficiently track raw materials, manage production workflows, and monitor sales and expenses. As a web-based application, Craftybase enables bakeries to maintain a real-time inventory of ingredients, finished products, and production batches, ensuring accurate stock levels and cost calculations. The system's functionalities allow business owners to create a comprehensive database of materials, track product recipes, and maintain a detailed history of production and sales. Additionally, Craftybase helps businesses handle financial tasks such as cost analysis, pricing strategies, and compliance reporting. With its powerful tracking and reporting features, Craftybase

provides an all-in-one solution for small bakery businesses to streamline operations, optimize costs, and improve overall efficiency (https://craftybase.com).



source (https://craftybase.com**)** ***Figure 2: CraftyBase admin dashboard***

craftybase has the following advantages and disadvantages.

Advantages.

- **Comprehensive Inventory & Cost Tracking:** Craftybase excels in tracking raw materials, finished goods, and production costs, making it an ideal tool for bakeries. It automatically deducts ingredient quantities when a batch is made, ensuring accurate stock levels and preventing overuse or shortages. Additionally, the platform calculates real-time cost per product, factoring in material costs, labor, and overhead, helping bakery owners set profitable pricing strategies. Unlike manual spreadsheets, which can be prone to human error, Craftybase provides automation and precision in inventory management.
- **Seamless Integration with E-commerce Platforms:** Craftybase integrates smoothly with popular online selling platforms like Shopify, Etsy, and Amazon. This is highly beneficial for

bakeries that sell online, as it enables automatic syncing of orders, inventory, and sales data. Instead of manually updating stock levels after each sale, Craftybase ensures that inventory is adjusted in real-time across all connected platforms. This integration reduces administrative workload, minimizes errors, and provides better visibility into product demand trends.

Disadvantages.

- **Subscription-Based Cost:** Unlike some open-source or free inventory management tools, Craftybase operates on a paid subscription model. This may be a drawback for small or newly established bakeries with limited budgets. While it provides valuable features, the monthly cost may not be justifiable for businesses that do not require advanced inventory tracking or e-commerce integrations. Some small bakeries might prefer simpler, free alternatives like spreadsheets or basic accounting software to manage their inventory.
- **Steep Learning Curve for Beginners:** Craftybase offers powerful features, but its extensive tracking system can be overwhelming for users who are new to inventory and production management software. Navigating through the platform, setting up recipes, and configuring cost tracking requires some learning. Unlike simpler inventory management tools, users may need time to fully understand how to maximize Craftybase's capabilities. This can lead to a slower adoption process, particularly for small bakery owners who may not have prior experience with such systems.

CHAPTER THREE:

RESEARCH METHOD A ND INTERNSHIP ACTIVITIES

INTRODUCTION

In order to guarantee the success and effectiveness of our project, it is necessary at this stage of our work to define precisely the methods adopted to develop the solution. This includes the modeling and enumeration of the various services that our system is supposed to offer to various users and the study of the various materials to be used**.**

3.1 CHOOSEN ENGINEERING METHOD

When developing software, we may choose any one of the following software engineering processes: waterfall model, V-Shape model, prototyping model, RAD, incremental, iterative, spiral, Agile. The choice of a model generally depends on the following factors: requirements, the date lines, risk factors, availability of technical resources, and financial resources: we choose the prototyping model to develop our system: in this section we first describe the prototyping model before justifying our choice;

3.1.1 Prototyping model

The Prototype Model is a Software Development approach useful for projects with vague or changing requirements. It involves creating a preliminary version of the software product, called a prototype, that shows the basic features and functionality that the customers want.

The prototype is then tested and improved by the customers and the developers until the customers are happy with the final prototype, which forms the basis for the final product. The Prototype Model has some benefits, such as increasing customer involvement, reducing risks, and improving quality.

However, it also has some drawbacks, such as being time-consuming, costly, and hard to document (The Knowledge Academy, n.d.-a).

A. TYPES OF PROTOTYPING MODEL

Prototyping Models are Software Development approaches that involve creating preliminary versions of the software product, called prototypes, that demonstrate the basic features and functionality that the customers want. There are four types of Prototyping Models:

1) Rapid Throwaway Prototyping

This technique offers a valuable method of exploring ideas and getting customer feedback for each of them. In this method, a developed prototype does not necessarily need to be a part of the ultimately accepted prototype. Customer feedback helps prevent unnecessary design faults, so the final prototype developed is of better quality.

The Rapid Throwaway Prototyping technique suits projects with unclear or changing requirements or when the developers want to experiment with different alternatives. The main advantage of this technique is that it allows the developers to identify and eliminate the flaws in the design quickly. The main disadvantage is that it can be wasteful and time-consuming, as the prototype is discarded after each iteration.

2) Evolutionary Prototyping

In this method, the prototype developed initially is incrementally refined based on customer feedback till it finally gets accepted. Compared to Rapid Throwaway Prototyping, Evolutionary Prototyping offers a better approach that saves time and effort. This is because developing a prototype from scratch for every iteration of the process can sometimes frustrate developers.

The Evolutionary Prototyping technique is suitable for projects with stable and well-defined requirements or when the developers want to deliver a working product as soon as possible. The main advantage of this technique is that it allows the developers to improve the quality and functionality of the product over time. The main disadvantage is that it can be challenging to document and maintain as the prototype undergoes

3) Incremental Prototyping

In this type of Incremental Prototyping, the final expected product is broken into different small pieces of prototypes and developed individually. Ultimately, when all individual pieces are appropriately developed, the various prototypes are collectively merged into a single final product in their predefined order. It's a very efficient approach that reduces the complexity of the development process, where the goal is divided into sub-parts, and each sub-part is developed individually.

The time interval between the projects's beginning and final delivery is substantially reduced because all system parts are prototyped and tested simultaneously. Of course, there might be the possibility that the pieces just do not fit together due to some lack of ness in the development phase this can only be fixed by careful and complete plotting of the entire system before prototyping starts.

The Incremental technique is suitable for projects with modular and independent components or when the developers want to deliver a functional product in stages. The main advantage of this technique is that it allows developers to focus on one part of the system at a time and get customer feedback. The main disadvantage is that it requires careful and complete planning of the entire system before prototyping starts, as any changes in the requirements or design can affect the integration of the prototypes.

4) Extreme Prototyping

This method is mainly used for Web Development. It consists of three sequential independent phases:

a) A basic prototype with all the existing static pages is presented in HTML format in this phase. This phase is used to create the user interface and navigation of the web application and get feedback from the customers. It allows the developers to design and test the layout and appearance of the web pages quickly. However, a basic prototype does not include any functionality or data processing of the web application**.**

b) In the second phase, functional screens are made with a simulated data process using a prototype services layer. This phase is used to create the web application's functionality and logic and get customer feedback. The phase allows the developers to quickly implement and test the features and behavior of the web application, although without any actual data or services of the web application. (The Knowledge Academy, n.d.-a)

c) This is the final step, where all the services are implemented and associated with the final prototype. This phase is used to create the data and services of the web application and integrate them with the user interface and functionality.

This Extreme Prototyping method makes the project cycling and delivery robust and fast. Moreover, it keeps the entire developer team focused and centralized on product deliveries rather than discovering all possible needs and specifications and adding necessitated features. This technique is suitable for projects with web-based and dynamic applications or when the developers want to use an agile and iterative approach to Web Development. The main advantage of this technique is that it allows

the developers to rapidly prototype and test the web application and get customer feedback. The main disadvantage is that implementing and integrating all the services with the final prototype can be complex and risky.

B. PHASES OF THE PROTOTYPING MODEL

The Prototype Model is characterized by its iterative nature, which allows for the development of a functional prototype followed by continuous refinement based on feedback.

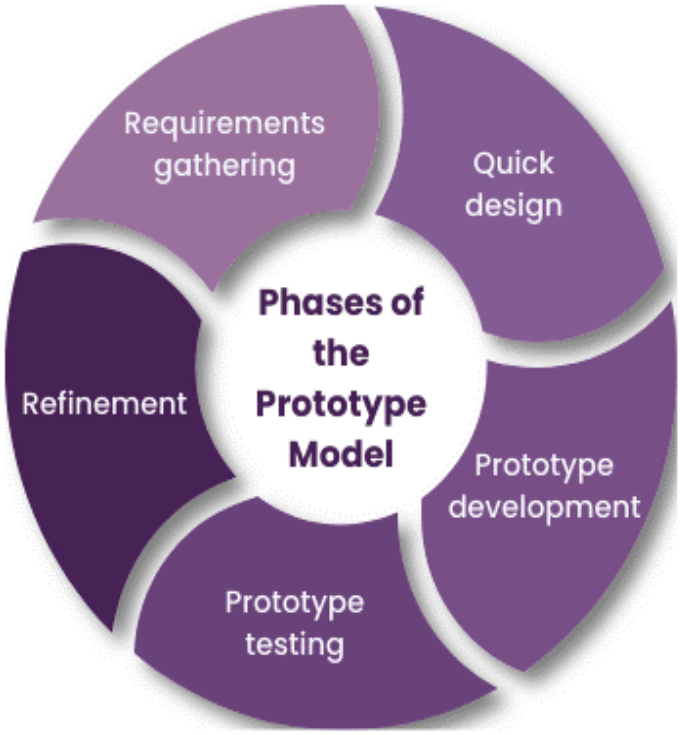


Figure 3 ***CraftyBase admin dashboard functionalities*** **Source (The Knoledge Academy, a) n.d.-)**

1. Analysis and Requirements gathering: During this phase, developers engage in in-depth discussions with stakeholders to identify the primary objectives of the software. However, unlike traditional methods, where all requirements are gathered up front, the Prototype Model focuses on capturing the essential features and functionalities that will be incorporated into the initial prototype.

2) Quick design: With the initial requirements in hand, developers create a quick and simplified design of the software's user interface and core functionalities. This design serves as a blueprint for the prototype, outlining the basic layout, user interactions, and essential features. The goal is to establish a foundation that can be built upon in subsequent phases.

3) Prototype development: Based on the quick design, developers begin creating the actual prototype of the software. This prototype is a working model that focuses on demonstrating the core features and functionalities of the software. It might need some of the intricate details and advanced features that would be present in the final product.

4) Prototype testing: Once the prototype is developed, it undergoes testing to identify any issues, bugs, or areas of improvement. This testing phase involves both developers and clients working collaboratively to evaluate the prototype's functionality and user experience. Clients interact with the prototype, providing real-time feedback on its usability and alignment with their requirements.

5) Refinement: Feedback obtained from prototype testing forms the basis for the refinement phase. Developers analyze the feedback and begin making necessary changes and improvements to the prototype.

These changes involve addressing identified issues, adding missing functionalities, and enhancing the user interface. The prototype is iteratively (The Knowledge Academy, n.d.-a) refined based on client feedback, ensuring that it becomes a more accurate representation of the final product with each iteration.

C. ADVANATGES OF THE PROTOTYPING MODEL

- The Prototype Model in Software Engineering offers several distinct advantages. It promotes enhanced collaboration between developers and clients, promoting active involvement and feedback throughout development.
- Early detection of issues is facilitated by creating a functional prototype, enabling timely identification and resolution of potential problems. The model's inherent flexibility and iterative approach accommodate changing requirements and allow for continual improvements, ensuring that the final product aligns more accurately with client expectations.
- The Prototype Model facilitates a deeper understanding of project requirements as developers and clients gain more precise insights into the software's functionalities and design, resulting in a more refined and effective product.

D. DISADVANTAGES

- The flexibility that allows for iterative improvements can also lead to scope creep, where evolving client demands may extend project timelines and increase complexity. Incomplete initial requirements might hinder the effectiveness of the prototype, requiring additional iterations to address client needs fully.
- There's a risk of unrealistic expectations, as clients might need to understand the prototype as the final product, potentially leading to disappointment if outcomes differ from prototype functionality or design.
- The iterative nature of the model can sometimes lead to time and resource constraints, as multiple refinement cycles may consume more resources than initially planned.

3.1.2 Justification of choice

- Digital software prototyping is more than just an average 3D model of your product. It mimics operation, functionality and allows testing of the final product before physically constructing it. By building a working simulation of the actual application, designers can effectively visualize, optimize and iterate the product digitally, saving time and money.
- Procrastination can badly affect businesses, especially in complex work environments where one has to meet tight deadlines. Digital prototyping enables quick action and quality decision making. Designers can start immediately, choosing from a number of prototyping tool available. Available, making changes and incorporating new requirements and ideas with ease.
- Digital prototyping enables users to continuously amend an agile model to reach the intended product design. It allows changes and refinements in quick succession, throughout the process. Using this methodology, designers can plan, optimize, validate, and visualize products beforehand without physically developing the actual product. This allows early identification and correction of faults in the product design. This technology also allows testing the model for robust qualities by subjecting it to load conditions, as a means to assess the product's behavior and interaction.
- Cost savings is one of the main driving forces behind incorporating digital software prototyping. Prototyping helps in minimalizing actual production time, thereby shortening the development cycle. It enables concurrent crafting among team members too, reducing the resources and labor required. Prior to digital prototyping, software engineers had to reach the end of the development phase before testing their designs. A single glitch could in extremis lead to the entire cycle being repeated incurring exaggerated and unnecessary costs. Today, design engineers can deploy digital software prototyping skills and recreate functional models, without incurring significant additional expenditure.

3.1.3 APPLICATION OF THE PROTOTYPING MODEL

Though we chose the prototyping model, we specifically chose the rapid Throwaway Prototyping model.

1. Analysis and Requirement gathering

To design this system**,** it was essential for us to collect the necessary information from specialists in the field. After structuring the information collected, we noticed that we needed the following things:

- A database for all the data related to the production of bread
- Recording of the different production tasks
- Inserting new bread offers into the database
- Inserting new employees into the database
- Inserting new orders from the customers into the database
- Providing the inventory for all the requirements needed for production on a particular night
- Statistical view of the different sales status.
- Statistics show the different financial objectives of the bakery.

Before talking about actual operation of the system, it is necessary to first define the functionalities that will be implemented within the system. So, this step will describe what we expect from our application. Then all of this will be modeled as a diagram using UML modeling language.

Management structure

- ❖ **Description:** An administrator records the production activity and records the employees' offer and orders in the system.
- ❖ **Modules**
  - **Inventory management**
  - **Order management**
  - **Employee management**
  - **Offer management**

Order request management

➢ **Description:** A user will store information's on his request for a particular unit of offer

➢ **Sub-modules**

- **Management of orders**
- **Production inventory**

Maintenance management

After identifying the requirements, we came out with some UML diagrams to model the requirements

3. Prototype development

Here we started implementing the back end of our features of our graphical user interface that we designed for it to operate its basic operations. We made use of algorithms to ease our work. This is the phase where most of the coding took part. This is where most of the coding and development took place.

Development technologies

Material (hardware) used for development

To carry out our work, we had at our disposal**:**

- A Lenovo ThinkPad T480s
- A Windows 11 professional operating system (64 bit)
- Intel(R) Core(TM) i5-8350U CPU @ 1.70GHz 1.90 GHz16.00 GB RAM memory
- A 320 GB 540 RPM SATA hard drive
- Software used for development
- vs.code IDE(version 1.48)

Client side

1) **Framework**

- LARAVEL 11
- Bootstrap CSS library to facilitate the design of our web pages and validation
- REACT NATIVE

2) **Languages**

- PHP: to create our migrations, controllers and models
- HTML 5: for formatting web pages
- CSS 3: for the style or our web components
- JavaScript: for client-side interaction

3) **Database server**

On the web server side

- PHP: it is a server scripting language
- My SQL: As our database

D. Prototype testing

Testing which is the process of evaluating and verifying that a software product or application does what it is supposed to do. And for this case it was tested through browsers by some Workers in the domain.

E. Refinement

After testing, the feedbacks came and we had to change things like the login page, the authentication method, the routing, and the different privileges to the different users.

3.2 INTERNSHIP ACTIVITIES

During this period of time, we did many activities during the internship even if some of them were far ahead of our level of academic hence I gained a great amount of knowledge and experience

3.2.0 Internship

An internship is a practical learning experience that bridges the gap between academic knowledge and real-world work environments. It provides an opportunity to apply theoretical concepts, gain hands-on experience, and develop essential professional skills. Internships allow students to work under industry experts, understand workplace dynamics, and enhance their problem-solving abilities. This experience plays a crucial role in career development, helping interns build confidence, improve technical expertise, and prepare for future job opportunities.

3.2.1 Activities carried out at CIA Consulting

During my internship at CIA Consulting, I actively contributed to the development of a dictionary application designed for the translation and learning of Ghomala, a local language. The project was built using Laravel for backend development and Bootstrap for frontend design, ensuring a responsive and user-friendly interface.

Key activities included:

Week 1: Project Setup & Requirement Analysis

- Set up the development environment (Laravel, Bootstrap, database).
- Understood project requirements and defined system architecture.
- Explored Laravel's Eloquent ORM and Bootstrap UI framework.

Week 2: Backend Development (Core Functionalities)

- Implemented authentication system (user registration & login).
- Developed database models and migrations for storing words and translations.
- Created controllers and routes for handling requests.

Week 3: Frontend Development (UI & UX Design)

- Designed a clean and user-friendly interface using Bootstrap.
- Ensured mobile responsiveness for different screen sizes.
- Integrated frontend components with backend APIs.

Week 4: Database Structuring & Optimization

- Structured the database to efficiently store and manage words, translations, and definitions.
- Applied relational database concepts to maintain data integrity.
- Optimized queries for faster data retrieval.

Week 5: Implementation of Search & Filtering Features

- Developed an advanced search mechanism for easy word lookup.
- Implemented filtering options based on category or pronunciation.
- Enhanced the search accuracy for better user experience.

Week 6: Testing & Debugging

- Conducted unit and integration testing to ensure functionality.
- Identified and fixed bugs affecting performance.
- Tested UI responsiveness across various devices and browsers.

Week 7: Team Collaboration & Version Control

- Worked closely with other developers to integrate features.
- Used Git and GitLab for version control and collaboration.

- Reviewed code, merged changes, and resolved conflicts.

Week 8: Documentation & Finalization

- Documented system features, functionalities, and database structure.
- Provided recommendations for future improvements.
- Presented final updates and wrapped up project deliverables

These activities enhanced my technical skills in web application development while contributing to language preservation through digital solutions. Let me know if you need modifications!

3.2.2 BENEFITS AND ACHIEVEMENTS DERIVED FROM THE INTERNSHIP

Some of the achievements I accomplished during my internship were:

1. **Enhanced Technical Skills**

- Gained hands-on experience with **Laravel** and **PHP**, which deepened my understanding of backend web development. Learning the **Eloquent ORM**, **routing**, and **Blade templating** in Laravel helped me build more efficient and scalable web applications.
- Improved **database management** skills, including creating efficient structures and queries to handle large sets of data, which is crucial for building applications that require real-time data retrieval and scalability.

2. **Practical Application of Web Development Tools**

- Developed a practical understanding of **Bootstrap** for frontend design, enabling me to build responsive and user-friendly interfaces for web applications.
- Gained experience in integrating frontend and backend systems, ensuring smooth communication between the user interface and the database.

3. **Problem-Solving and Critical Thinking**

- Faced with real-world challenges, I learned how to troubleshoot and solve issues, such as **optimizing database performance** and building a complex **search and filter mechanism**.
- Acquired the ability to approach problems methodically, break them down into manageable tasks, and work efficiently under time constraints.

4. **Contribution to Language Preservation**

- This contributed to the preservation and learning of **Ghomala**, a local language, by developing a dictionary application that facilitates language translation and learning. This provided me with a sense of fulfillment and helped me realize the societal impact of tech projects.

5. **Improved Collaboration and Communication Skills**

- Worked closely with a team of developers, learning to communicate effectively and contribute to collaborative problem-solving. This experience enhanced my teamwork and project management skills, especially when coordinating tasks and handling deadlines.

6. **Professional Growth and Networking**

- The internship provided exposure to the professional world, allowing me to interact with senior developers and gain insights into the tech industry. I expanded my network and gained valuable mentorship, which will help in my future career.

3.2.3 CHALLENGES DURING THE INTERNSHIP

Some of the challenges I faced during my internship were**:**

1. **Data Entry and JSON Object Creation**

- Inputting Ghomala words and their translations into JSON objects for database insertion.
- Ensuring accuracy and correct formatting to prevent errors during data import.
- Handling the large volume of words and translations, which was time-consuming and repetitive.

2. **Learning Laravel and PHP**

- Steep learning curve in mastering Laravel and PHP as a beginner.
- Understanding complex features like routing, middleware, Eloquent ORM, and Blade templating.
- Building efficient database queries and implement advanced features such as search and filtering.

3**. Database Structure and Performance Optimization**

- Designing an efficient database to manage a large volume of data while maintaining performance.
- Optimizing queries for fast retrieval of words and translations.
- Balancing data integrity and system performance.

3.3 Application of topic to internship

Object-Oriented Modeling (OOM) using UML

Object-Oriented Modeling (OOM) is a methodology used in software development to represent real-world entities as objects that encapsulate both data and behavior. It provides a structured approach to designing and analyzing a system using Object-Oriented Programming (OOP) principles such as encapsulation, inheritance, polymorphism, and abstraction.

Unified Modeling Language (UML) is the standard notation for object-oriented modeling, offering a set of diagrams to visualize, specify, and document software systems. For the **Bakery Management System**, UML helps in designing the system structure, interactions, and behaviors effectively before implementation.

Key UML diagrams used in the project include:

1. **Use Case Diagram** – Defines the interactions between users (such as bakery staff, customers, and administrators) and the system, highlighting the main functionalities like order processing, inventory management, and sales tracking.
2. **Class Diagram** – Represents the system's static structure by defining classes, attributes, methods, and relationships between them (e.g., Order, Customer, Product, Invoice).

3. **Sequence Diagram** – Illustrates the flow of interactions between objects in a specific scenario, such as order placement and payment processing.
4. **Activity Diagram** – Depicts the workflow of system operations, such as the process of managing stock levels or generating sales reports.

Using UML ensures a clear and structured design for the Bakery Management System, reducing development errors and improving maintainability. It provides a visual blueprint that enhances communication between developers and stakeholders, ensuring a smooth transition from design to implementation.

a) Use case diagrams

Use case diagram is a representation of a user's and administrator's interaction with the system that shows the Relationship between the user, administrators and the different use cases in which the user is involved.

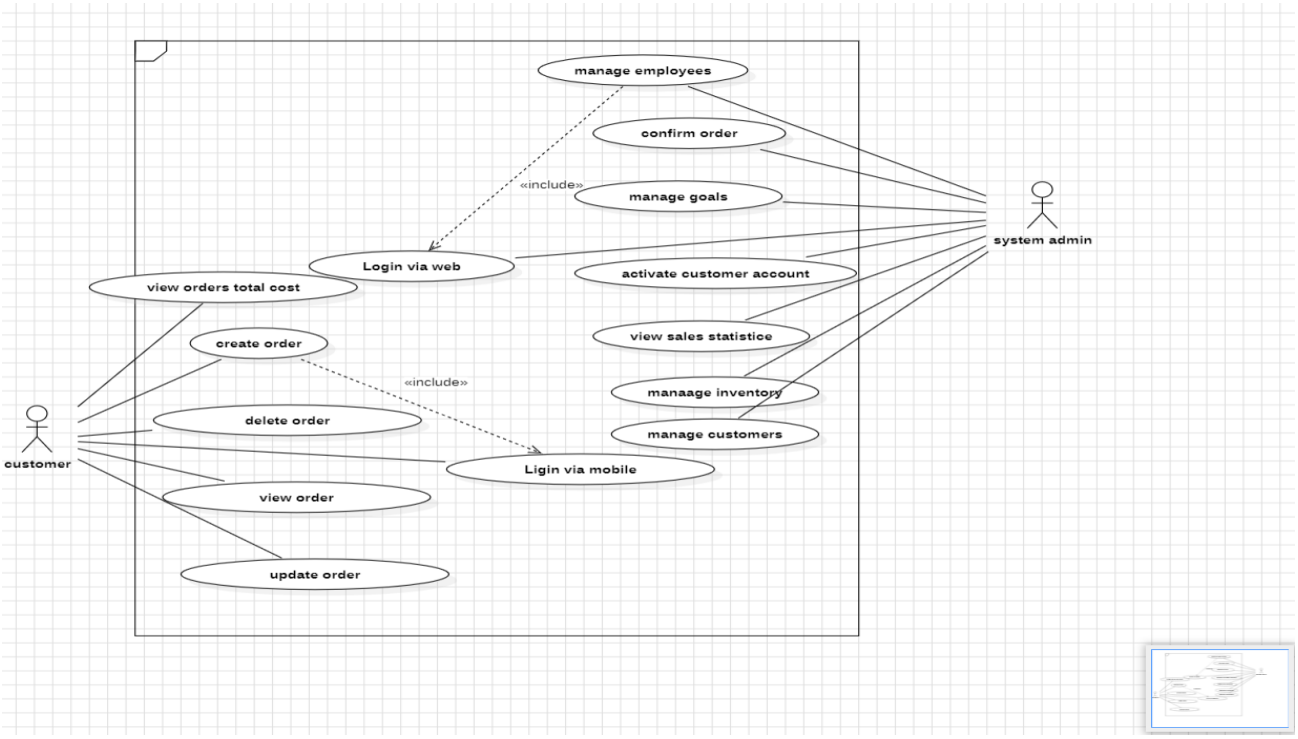


Figure 4 ***Use case diagrams***

Source (bakery_Ms UML Diagrams.mdj | Star UML)

b) Activity diagram

The description of the use case can contain several alternatives and /or exceptional scenarios. It is Therefore, difficult to have a vision of all the actions. The activity diagram is a graphical means to Give these overviews**.**

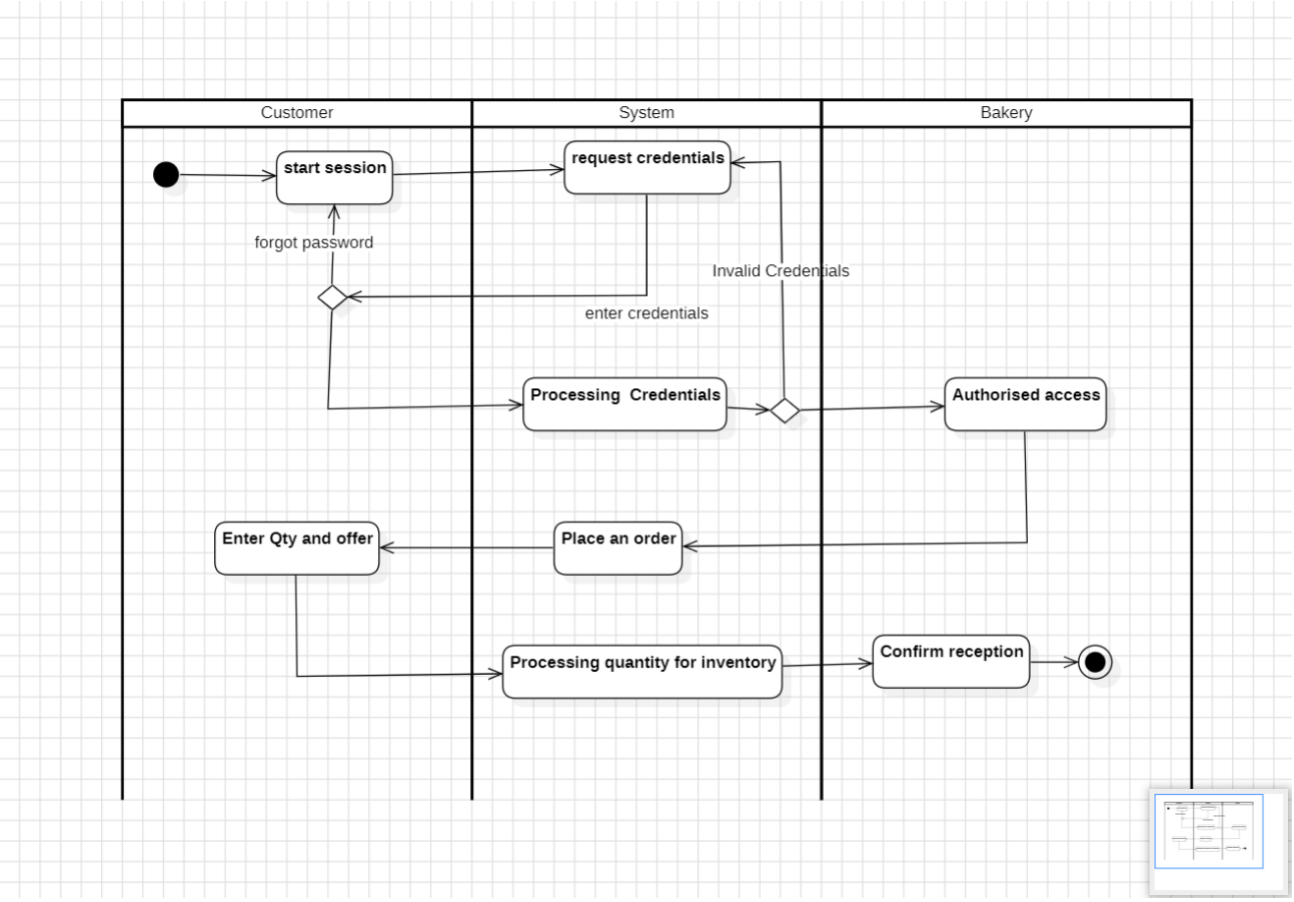


Figure 5 Activity diagram Bakery_Ms UML Diagrams.mdj | StarUML)

c) Sequence diagram

Sequence Diagrams are interaction diagrams that detail how operations are carried out. They capture the interaction between objects in the context of a collaboration. Sequence Diagrams are time

focus and they show the order of the interaction visually by using the vertical axis of the diagram to represent time what messages are sent and when.

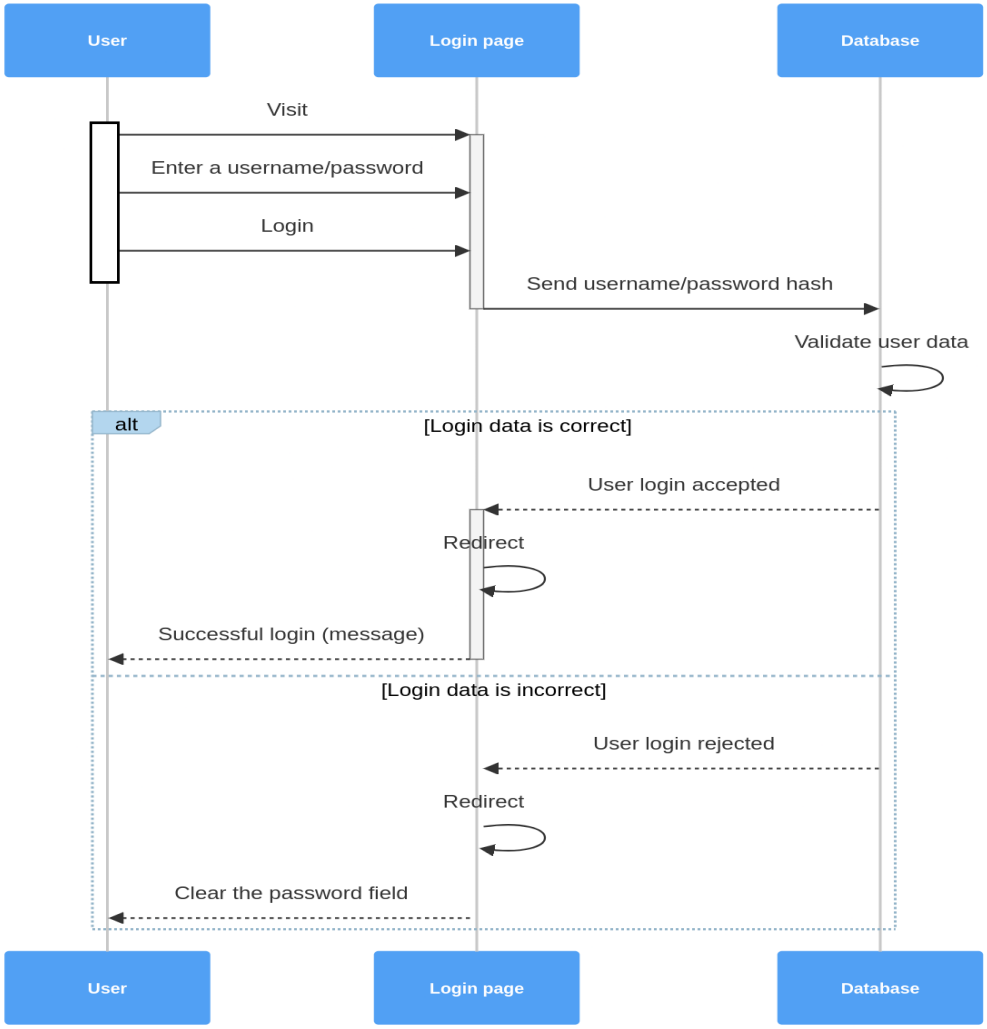


Figure 6 Sequence diagram ***Source (Untitled.Vpd | Visual Paradigm Online, n.d)***

The figure above represents a sequence diagram concerning the use case authentication. It permits all interaction existing between the actors to permit authentication.

d) Class diagram: A class diagram shows the internal structure of the system. It permits to deliver an abstract representation of the object of the system which will interact together to realize a use case.it is a static view because we don't consider the time factor in the behavior of the system. The class diagram models concept of the fields of the application in the context of implementation of an application

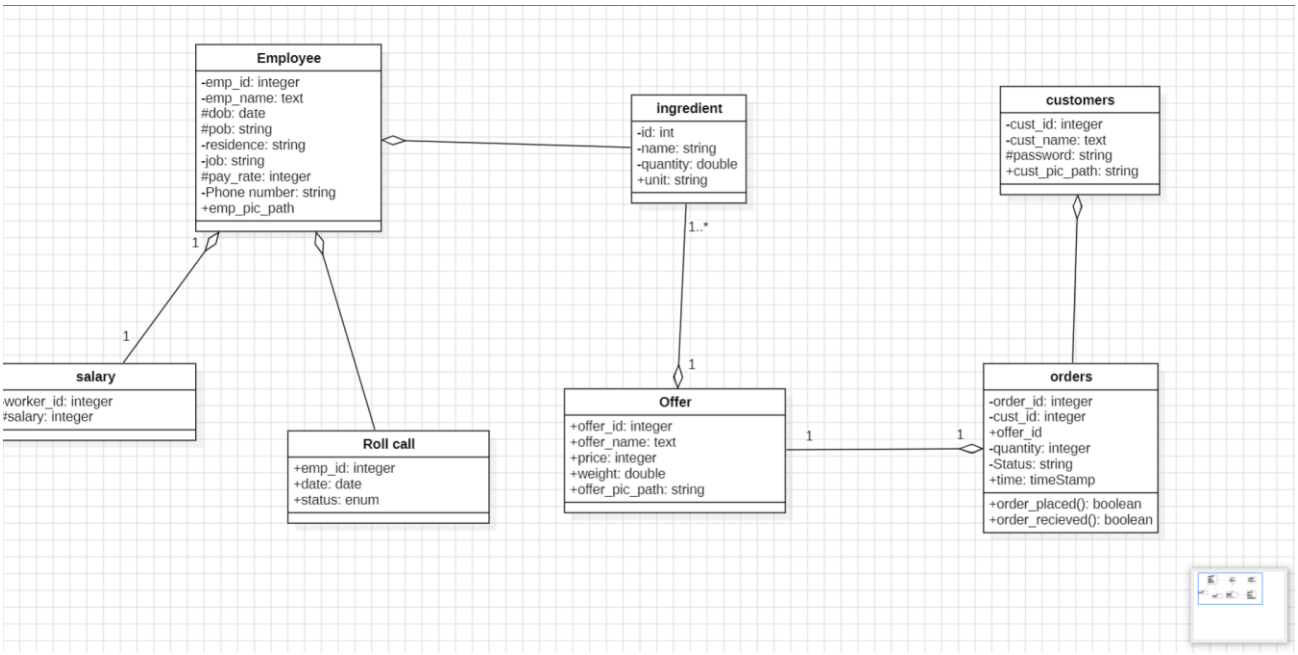


Figure 7 : Class diagram Source Bakery_Ms UML Diagrams.mdj | StarUML)

CHAPTER FOUR:

PRESENTATION, ANALYSIS AND INTERPRETATION OF RESULTS.

4.1 PRESENTATION OF ORGANISATION

4.1.1 HISTORICAL BACKGROUND OF CIA CONSULTING

The company was founded with the vision of providing innovative solutions to businesses and individuals facing technological and operational challenges. From its inception, it has focused on delivering high-quality services and empowering professionals through specialized training in emerging technologies.

Over the years, the company has expanded its expertise, adapting to industry trends and market demands. By collaborating with various partners and clients, it has successfully developed solutions that drive efficiency, productivity, and sustainable growth. Today, the company continues to innovate, offering cutting-edge tools and training programs to support businesses and professionals in achieving their goals.

4.1.2 ACTVITIES OF THE ORGANISATION

CIA Consulting provides innovative solutions to help clients and partners effectively address their challenges. Additionally, we train young professionals and businesses in modern technologies and techniques to accelerate their success. Their mission is to support individuals and organizations in their growth by delivering high-quality services and efficient solutions. We believe that sustainable and impactful solutions are essential for addressing real-world problems, contributing to positive change in communities and industries.

Key Activities:

- Development of tailored solutions for businesses and individuals.
- Training programs on emerging technologies and professional skills.
- Providing high-quality services to enhance efficiency and productivity.
- Contributing to sustainable and impactful solutions for long-term benefits.

4.2 Results

During the process of development, we developed 2 prototypes till date and that's the 2 prototype I'm going to present below**.**

4.2.2 First prototype

The first prototype was developed with react js and the design was mainly based on glasmorphysm display with a Laravel based rest API.

1.**Login page design**

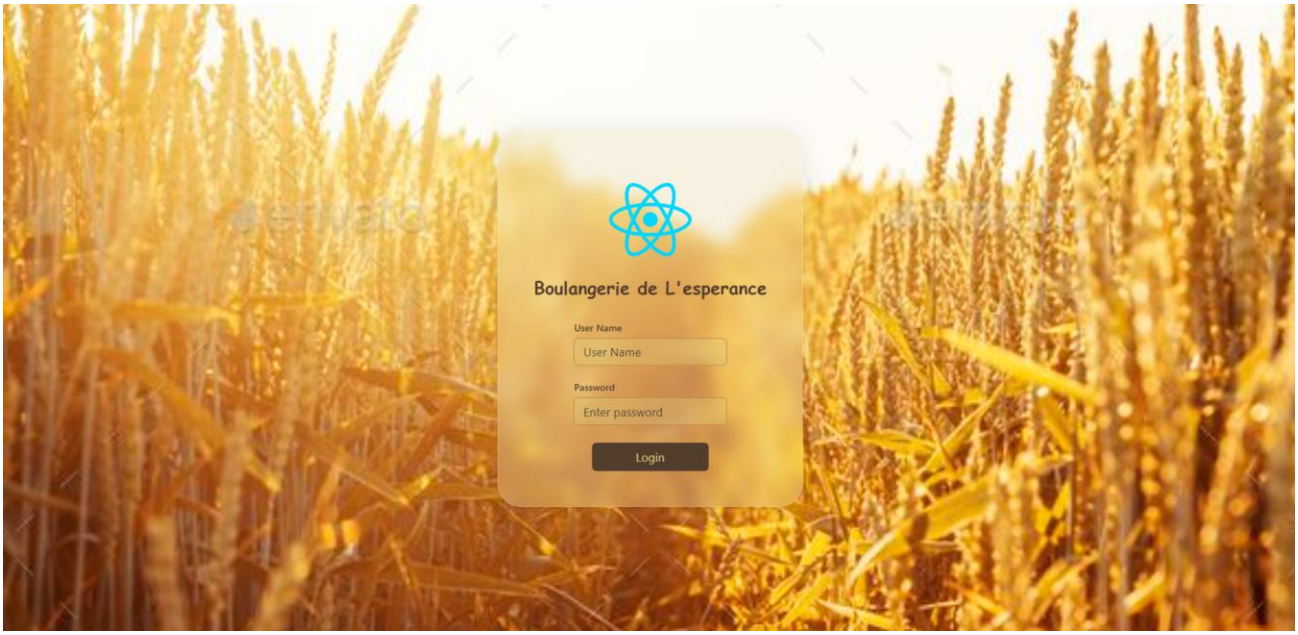


Figure 9 : login(prototype1)

4.2.3 Prototype 2

This second prototype was developed as a full Laravel web app with blade as the template generator with bootstrap 5 and jetstream for authentication. And also the implementation of a mobile app for the passing of orders by the different customers

1. Login page: Where the admin enters or logins into the system

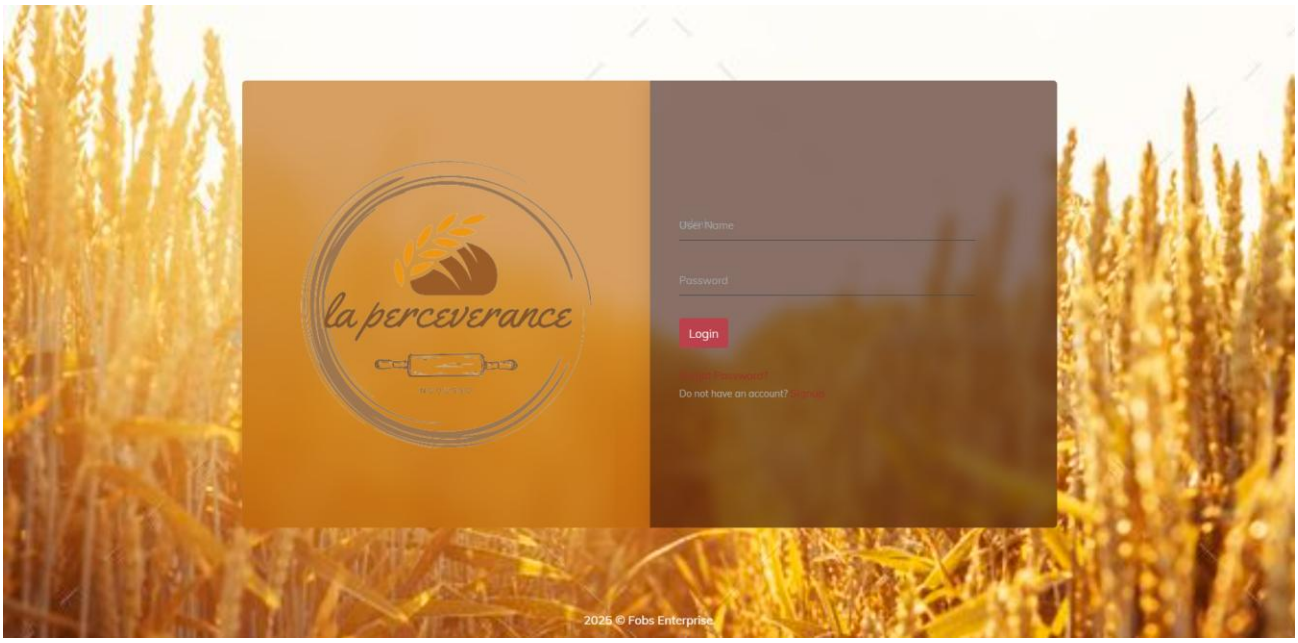


Figure 10 : login(prototype2)

1. Employee

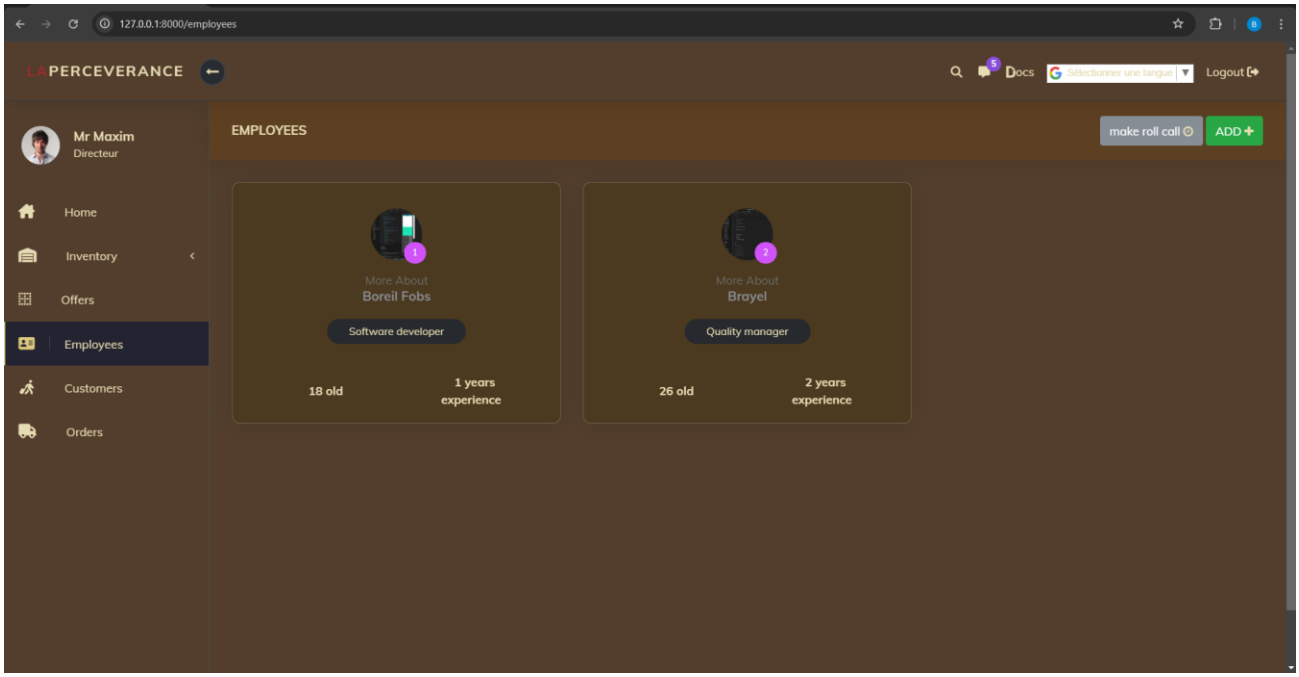


Figure 11 : Employee panel

where you can view, delete, add employee clicking on the card will bring you to see more about the employee with possibility to delete and update

2. Offers:

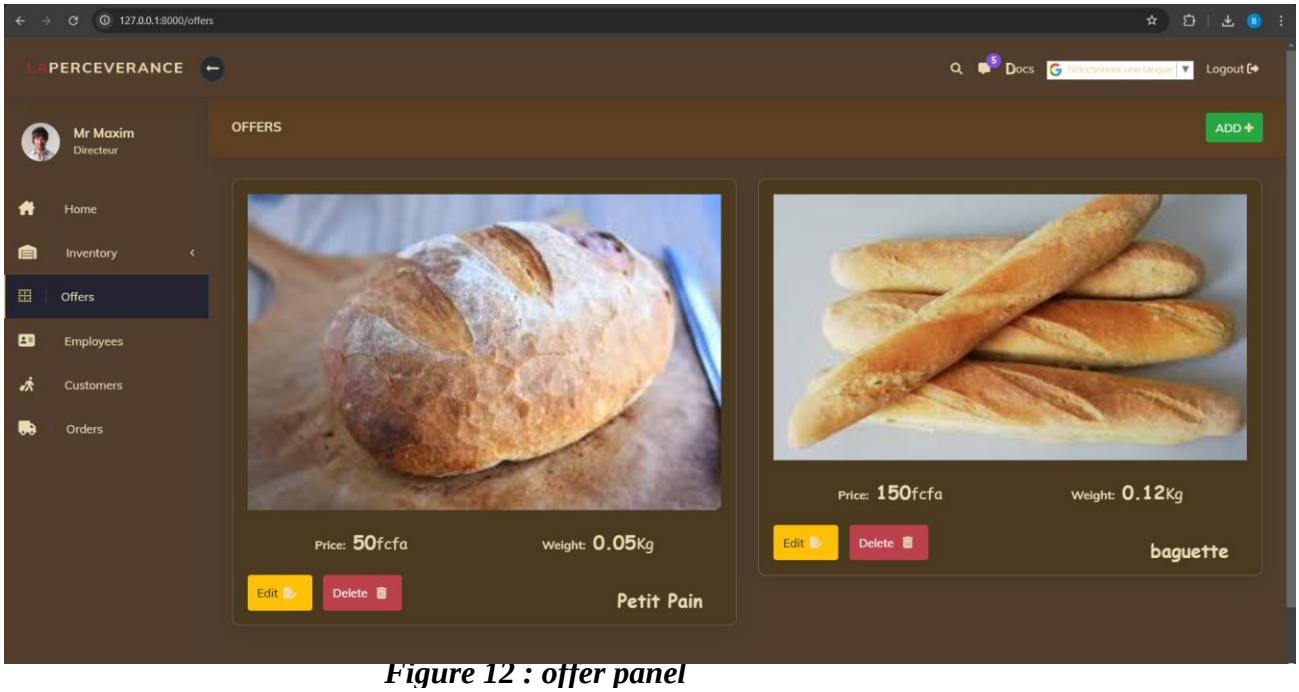


Figure 12 : offer panel

this is the page where the admin can create, update and delete a particular bread offer the mass shall be used later during inventory management.

3. Orders-web-view:

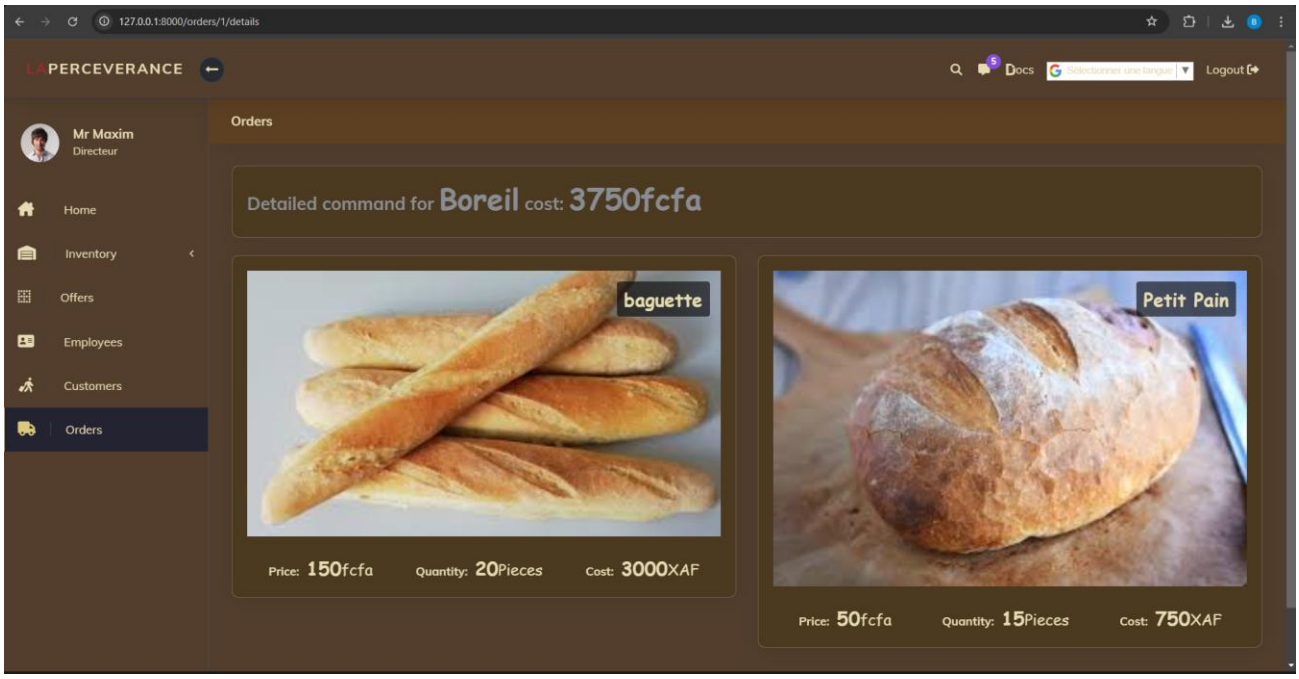


Figure 13 : detailed order

This is where the administrator will view the different orders for each customer and the total amount to play.

4. Ingredient

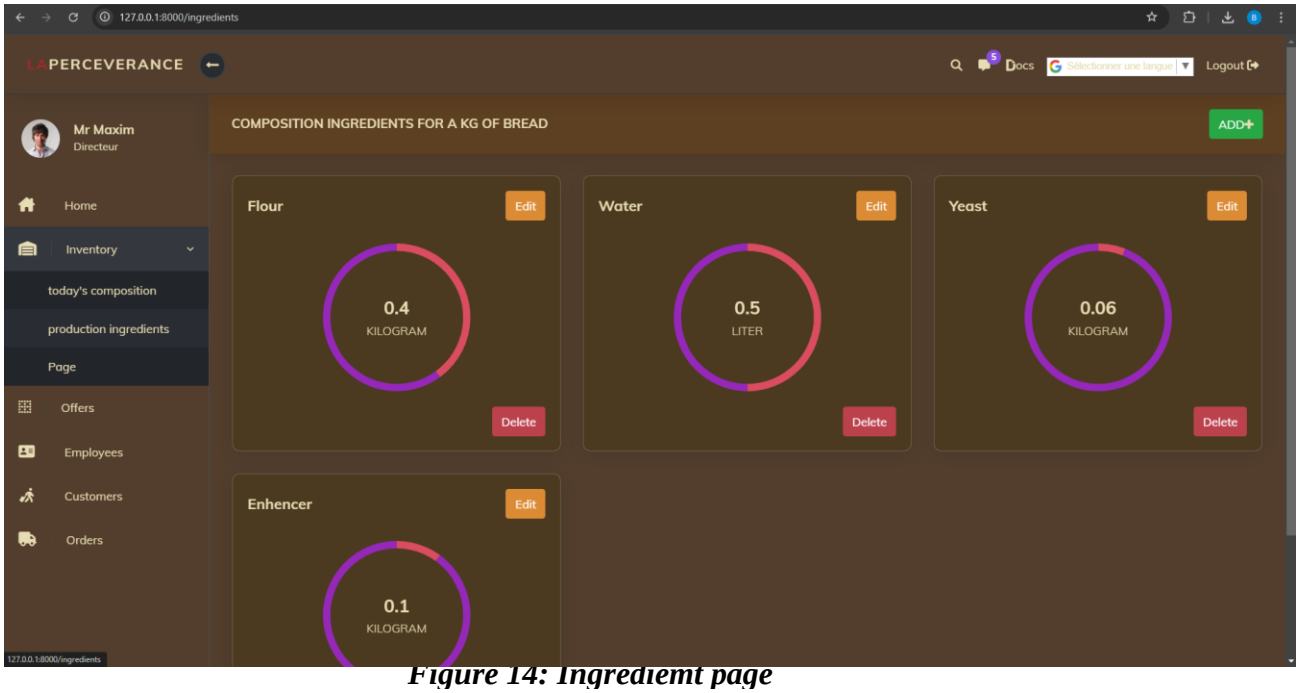
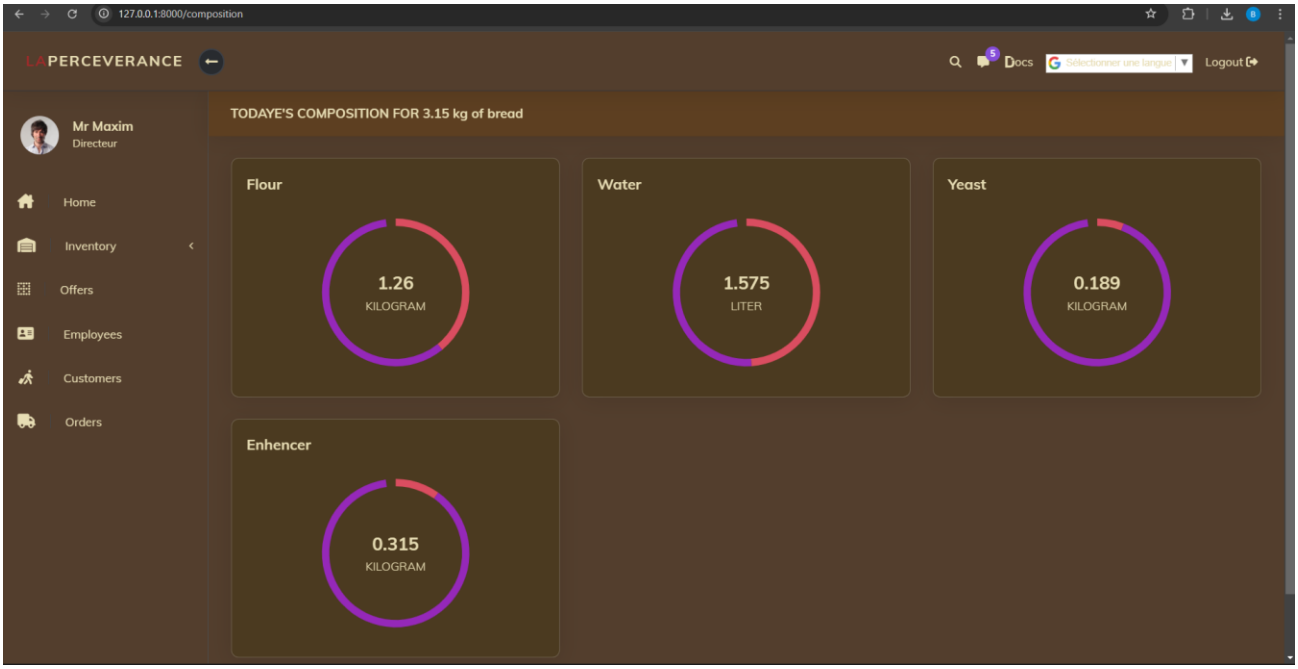


Figure 14: Ingrediemt page

This is the page where the admin will add, update, and delete all the ingredients needed for the production of 1kg mass of finished bread so it will be used for inventory. And below we see quantity of ingredients needed for a production of all orders to deliver which is the composition.



5. **Login and register for customer**

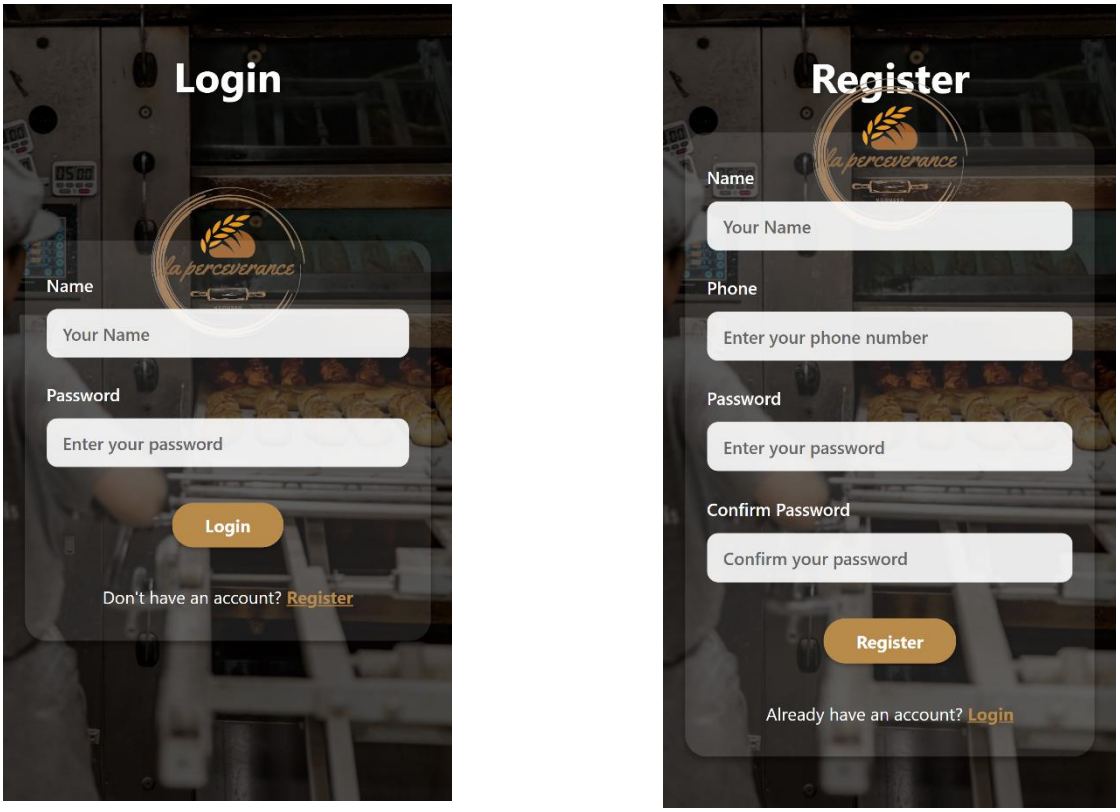


Figure 15 : Customer login/register

The pages where the customers will login and register to the bakeries platform.

6. **Home page and order form**

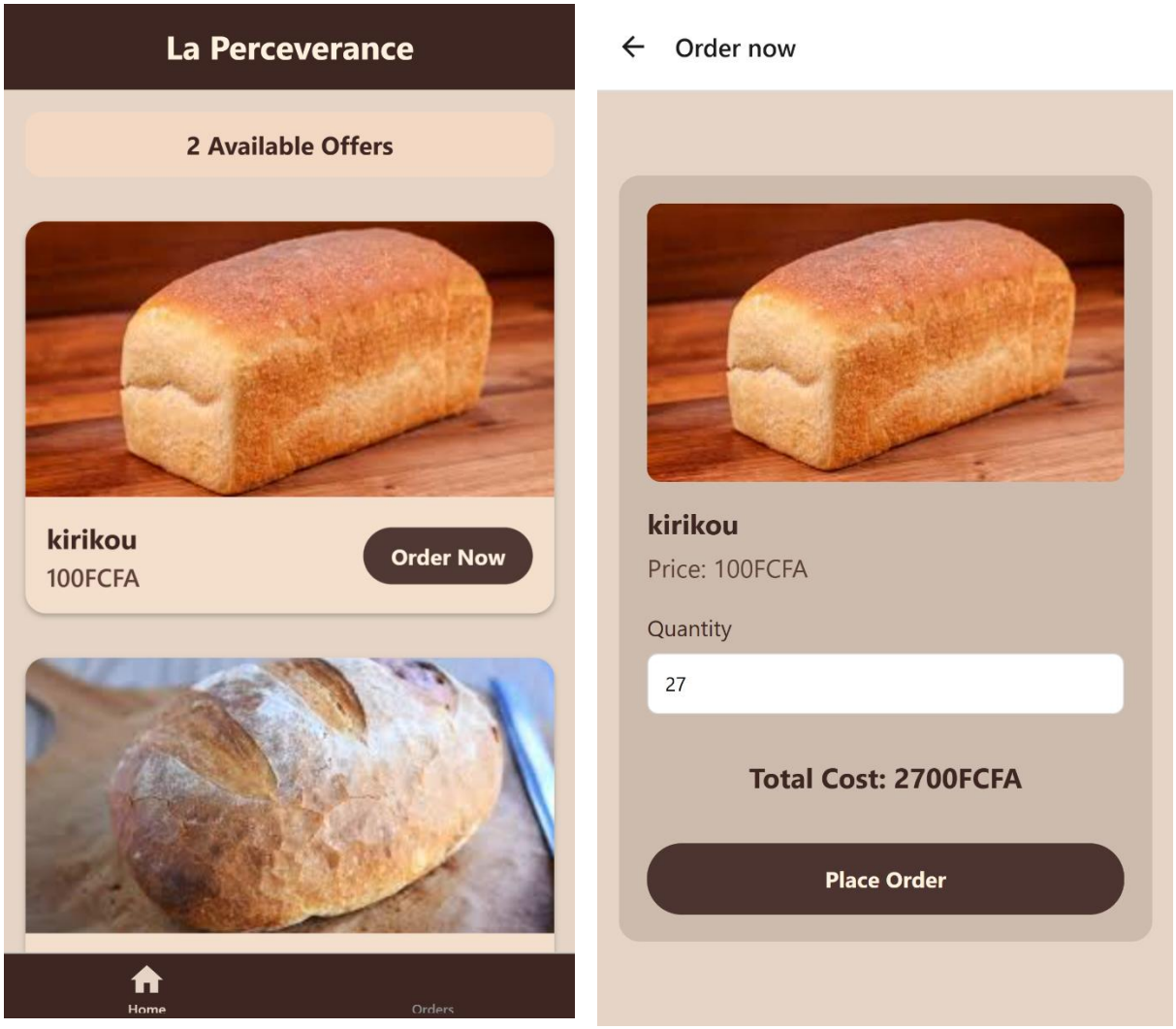


Figure 16: Customer home

This is where the customer will order his product which was created from the admin panel

7. Customers' Orders

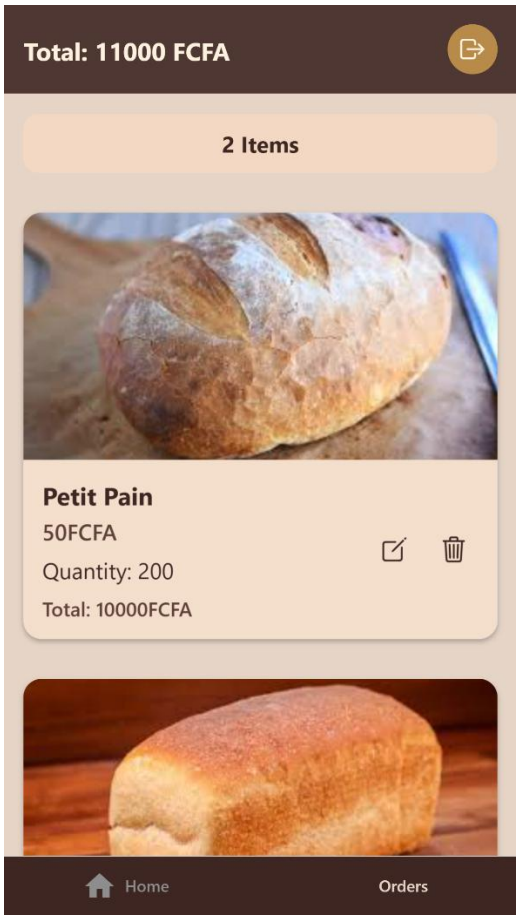


Figure 17 : customer orders

This is where the customer will view the different orders he placed with the ability to update and delete any of them.

CHAPTER FIVE:

SUMMARY OF FINDINGS AND RECOMMENDATIONS

INTRODUCTION

Our study focused on the creation of a web application to computerize the maintenance management process. To do this, we carried out a preliminary study on existing systems followed by the modeling of our system, modeling guided by the UML method through some of these diagrams, which allowed us to set up a system semi-functional for the hour, the progress of which is estimated at 93.5%. The first task we set up out to do was framing our project with the Project Director with whom we decided on certain major axes of the project such as the deployment environment, the technologies and languages to be used. We then had to agree to build the conceptual data model which is the core of the application and which would allow us to build our database from which the implementation would then really take on its full meaning.

5.1 Summary of findings

- **Excess Inventory Issues:** The bakery had a large amount of excess ingredients that were not being used efficiently, leading to wasted resources. Implementing real-time inventory tracking helped identify and address these inefficiencies.
- **Frequent Ingredient Shortages:** A common issue was the lack of sufficient raw materials due to poor inventory management. By integrating a real-time tracking system, the bakery was able to better predict and manage stock levels, reducing shortages and ensuring consistent production.
- **Limited Custom Software Solutions:** Many of the tools and processes used for bakery management were either outdated or purchased off-the-shelf, which did not fully cater to the bakery's unique needs. The development of a tailored bakery management system provided more relevant features like ingredient tracking, order management, and customer data integration.
- **Lack of Technology Knowledge Among Staff:** Some bakery staff were unfamiliar with recent technology and software tools. This was addressed by providing training on the new system to help staff improve their understanding and use of digital tools in day-to-day operations.

- **Manual Record-Keeping:** Prior to implementing the bakery management system, records of sales, inventory, and orders were kept manually. This often led to errors and inefficiencies. The new system automated these processes, improving accuracy and speed.

5.2 Recommendation

- **Staff Training**: CIA Consulting should invest in regular training programs to keep developers updated with the latest web development technologies, particularly Laravel and PHP.

8. **Standardized Documentation**: Implement a more efficient system for documenting technical tasks and project details to improve knowledge sharing and collaboration.
9. **Internship Development**: Encourage future interns to actively seek feedback and make use of learning opportunities to improve their skills and work performance.

5.3 Conclusion

In conclusion My internship at CIA COSULTING was a valuable and rewarding experience that helped me achieve my objectives and prepare for my future career. I gained practical skills and knowledge in developing innovative solutions, training methodologies, and the application of modern technologies to real-world problems. This experience allowed me to apply theoretical concepts from my studies, enhance my problem-solving abilities, and develop essential professional skills such as communication, teamwork, and adaptability. Additionally, I gained insights into the challenges and opportunities in the field of technology-driven solutions for businesses and individuals.

I would like to express my gratitude to CIA CONSULTING, the entire team, and especially my supervisor TOUBE PIERRE for their support and guidance throughout my internship. I appreciate the opportunity to contribute to meaningful projects and hope to continue collaborating with them in the future.

REFERENCES

Online Newspaper Articles:

Anshul_A. (2024, March 01). Introduction to visual studio -GeeksForGeeks. Retrieved from GeeksForGeeks.org: https://www.geeksforgeeks.org/introduction-to-visual-studio/

Windows update tips. (2024, January). Retrieved from stackoverflow: www.statckoverflow.com

("GLPi Review 2024," n.d.-a; 6 Reasons Why You Must Deploy Software Prototyping, n.d.-a; 6 Reasons Why You Must Deploy Software Prototyping, n.d.-a; TheKnowledgeAcademy, n.d.-a; "GLPi Review 2024," n.d.-b; "GLPi Review 2024," n.d.-c; "GLPi Review 2024," n.d.-d; GLPi Reviews - Pros & Cons, Ratings & More, n.d., n.d.-b, n.d.-c, n.d.-d; Altvater, 2017; Untitled.Vpd | Visual Paradigm Online, n.d.)

6 Reasons Why You Must Deploy Software Prototyping. (n.d.). Retrieved March 25, 2024, from https://www.rezaid.co.uk/post/6-reasons-why-you-must-deploy-software-prototyping

Altvater, A. (2017, April 7). Top Load Testing Tools: The Best Tools for Load Testing Reviewed & Compared. Stackify. https://stackify.com/top-load-testing-tools/

GLPi Review 2024: Features, Pros, Cons and Pricing. (n.d.-a). Tech.Co. Retrieved March 23, 2024, from https://tech.co/asset-tracking/glpi-review

GLPi Review 2024: Features, Pros, Cons and Pricing. (n.d.-b). Tech.Co. Retrieved March 25, 2024, from https://tech.co/asset-tracking/glpi-review

GLPi Review 2024: Features, Pros, Cons and Pricing. (n.d.-c). Tech.Co. Retrieved March 23, 2024, from https://tech.co/asset-tracking/glpi-review

GLPi Review 2024: Features, Pros, Cons and Pricing. (n.d.-d). Tech.Co. Retrieved March 25, 2024, from https://tech.co/asset-tracking/glpi-review

GLPi Reviews—Pros & Cons, Ratings & more. (n.d.). GetApp. Retrieved March 23, 2024, from https://www.getapp.com/it-management-software/a/glpi/reviews/

TheKnowledgeAcademy. (n.d.-a). Prototype Model In Software Engineering: All You Need To Know. Retrieved March 25, 2024, from https://www.theknowledgeacademy.com/blog/prototype-model-in-software-engineering/

TheKnowledgeAcademy. (n.d.-b). Prototype Model In Software Engineering: All You Need To Know. Retrieved March 25, 2024, from https://www.theknowledgeacademy.com/blog/prototype-model-in-software-engineering/

TheKnowledgeAcademy. (n.d.-c). Prototype Model In Software Engineering: All You Need To Know. Retrieved March 25, 2024, from https://www.theknowledgeacademy.com/blog/prototype-model-in-software-engineering/

TheKnowledgeAcademy. (n.d.-d). Prototype Model In Software Engineering: All You Need To Know. Retrieved March 25, 2024, from https://www.theknowledgeacademy.com/blog/prototype-model-in-software-engineering/

APPENDIX

Appendix A: Data set

DEPARTEMENT	Utilisateur	Marque	N0 Serie
DAD	KUMAR	Hp Laserjet P 2035	VNC4G42514
	KODOCK	Hp COLOR LASERJET PRO MFP 277	
	SALLE	CANON IR 2520	
SECURITY	BUREAU SECURITY MANAGI	Hp Deskjet 4530	TH6813D0ZQ
	OPERATING SECURITY	Hp Laserjet Pro MFP 227	VNC4509749
SUSTAINABILITY	AMBASSA ADELAIDE	Hp Deskjet 3000	CN1CE32 H5W
	AGATH	Hp LASERJET PRO M120 a	VNC6K06691
	GUINSOM ROMEO	HP	
	BUREAU ASS SOCIAL INTERNE	Hp Lserjet P2035	VNC4G23083
	MENGUE J	Hp Lserjet P2035	CNCF460717
	NJE NJE	Hp Laserjet M426 DW	PHB8JD10VL
AUDIT INTERNE	YMBNE	Hp Deskjet 3000	CN0AE11GP2
DDP	DDP	Hp Laserjet M806	JPDVL8DOJZ
MAGASIN	NJOH O	Hp Laserjet P2035	VNC4B29741
	TIOBO	Hp Laserjet P2035	VNC4G61153
	NDZANA R	Hp Laserjet P2035	VNC4F52297
	SALLE	CANON IR 2520	(21) RML87537
COMMUNICATION	ZOOBO J C	Hp Deskjet 3000	CN1CE32K2Z
DPE/MA	Ex RAJESH	Hp Deskjet 4645	CN530570XR
	SECRETARIAT	CANON IR 2202	
	SECRETARIAT	Hp Laserjet Pro MFP 227	VNC4G22731
	KWEM OSCAR	Hp Officejet Pro 8710	CN6AIH323C
	MINTANGA	Hp Laserjet Pro M402	PHCGD04543
	TSAFFO M	Hp Laserjet Pro M402	
DT	ATSANG L	Hp Laserjet Pro M402	PHC6C08359

Fiaure 18: dataset

Appendix B: Code Listing

Algorithm to delete offer from database

Algorithm ProductDelete (request, pk)

1. Begin
2. Check if user is logged in
3. if not, redirect to 'user-login'
4. Check if user role is 'Admin'
5. if not, access is denied
6. Get the offer with id 'pk'
7. If the request method is 'POST'
8. Delete the product
9. Redirect to offers
10. Else
11. Prepare the context with 'item' as the product
12. Render 'dashboard/products_delete.html' with the context
13. End if
14. End

Function to create offer to the database

```
public function store(Request $request){


    $request->validate([

      'name' => 'required',
```

```
        'price' => 'required|numeric',
        'mass' => 'required|numeric',
        'offer_pic' => 'required|max:2048',
    ]);

    // handle image upload

     if($request->file('offer_pic')){
        $img_path = $request->file('offer_pic')->store('offer_pics', 'public');
        $img_url = Storage::url($img_path);
    }

    if($request->file('offer_pic')){
        Offer::create([
            'name' => $request->name,
            'price' => $request->price,
            'mass' => $request->mass,
            'offer_pic_path' => $img_url
        ]);
    }else{
         Offer::create([
            'name' => $request->name,
```

```
            'price' => $request->price,
            'mass' => $request->mass,
        ]);
    }
    return redirect('/offers')->with('success', 'offer created successfully');
}
```

Appendix C: statically analysis and graphical representations

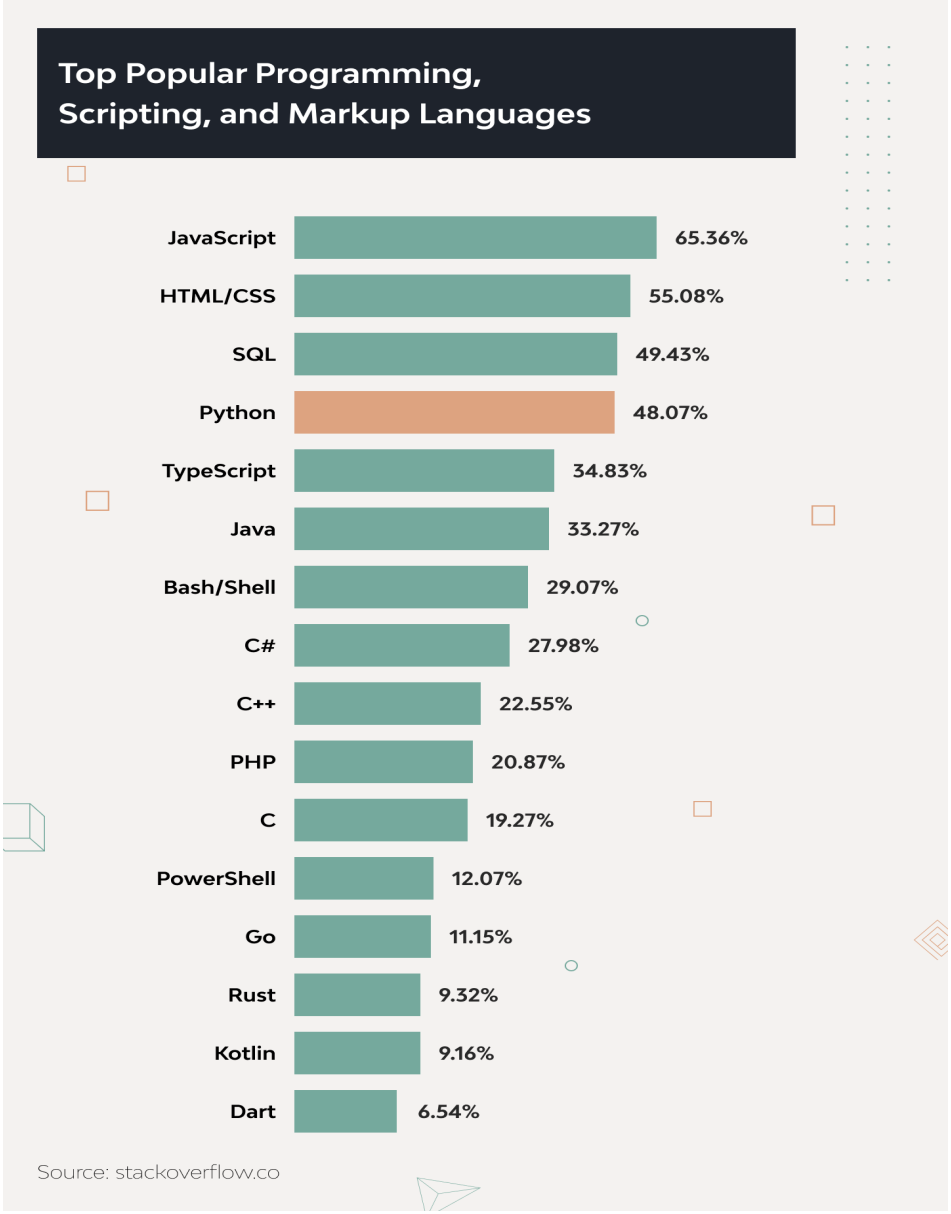


Figure 19: popular languages **Source (The Knowledge Academy, n.d.-b)**

Appendix D: Certificate of completion of the internship

CIA CONSULTING
Formation & Consulting
Expertise dans le domaine informatique
CAMEROUN, Yaoundé, Rue 4 111, Anguissa
(+237) 692 17 50 99 / 680 55 07 75
www.cia-service.com

CIA CONSULTING
Information Technology Specialist
Expertise dans le domaine informatique
CAMEROON, Yaounde, Street 4 111, Anguissa
(+237) 692 17 50 99 / 680 55 07 75
www.cia-service.com

ATTESTATION

DE FIN DE STAGE

Par la présente, nous attestons que,

M. FOBASSO TOUOTSOP Boreil

a effectué au sein de notre structure, un stage de 02 mois dans le domaine du **Développement d'Applications** pendant la période allant **du 08/07/2024 au 06/09/2024.**

Durant cette période, il a démontré un réel intérêt et une grande motivation pour les tâches qui lui ont été confiées par ses encadreurs.

En foi de quoi, la présente attestation est délivrée pour servir et faire valoir ce que de droit.

Fait à Yaoundé, le 06 SEPT 2024

Le Directeur Général

Directeur Général
GENERAL MANAGER

Figure 20 Certificate of completion of the internship

Appendix E: prototype

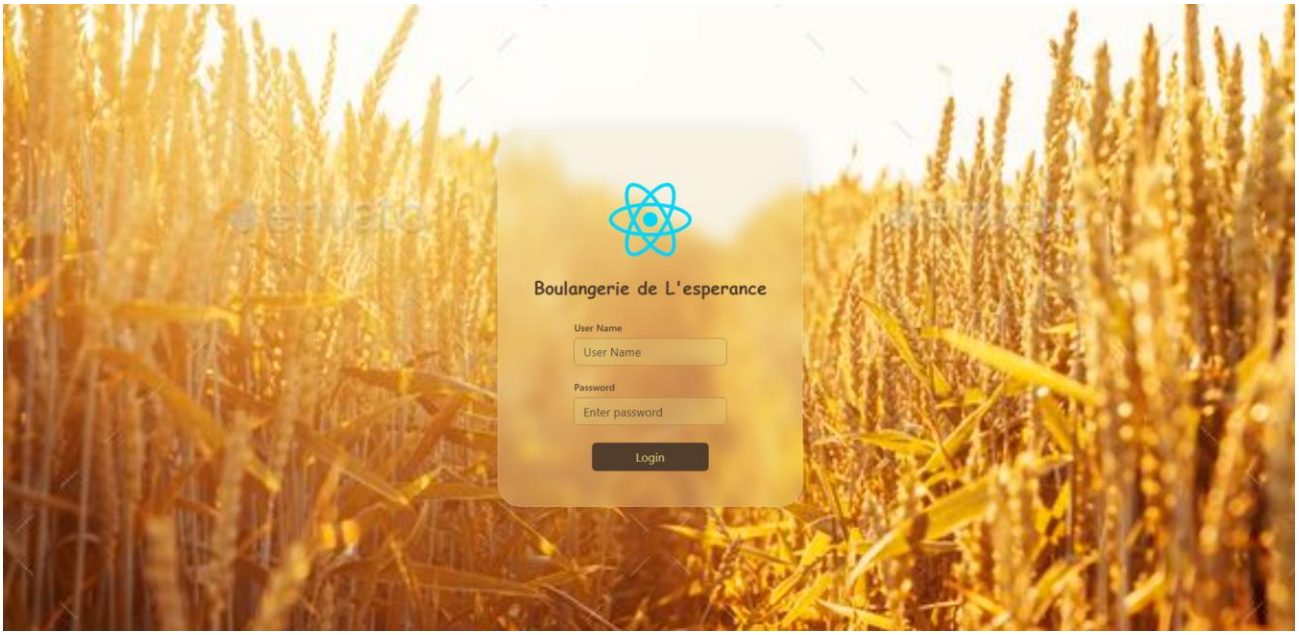


Figure 21 prototype 1