CHAPTER 1: INTRODUCTION

Introduction

The project Online Bike Rental System is the complete web-based design on Laravel using Visual Studio Code Editor Software. The aim of the project is to develop Online Bike Rental System Model Software in which all the information regarding the Rent bike available in the organization will be presented. By using this system users can book the bike as rent for sutten days. Then Admin will check the rent details of the user and approve or reject it. It is an internet-based web application that has an admin component to manage bikes in the system. This web application is based on the management of rented bikes in the organization. The application contains Renting Bike Catalog, Rent Details, Reports, and other essential Bike Rental System entities. There is a provision for updating the bike information also. This application also provides information on available bikes and rentals as well as transaction details. Each new bike is registered according to its properties. Here Login page and Register page are created in order to implement authorization and authentication of the system users.

Statement of Problem

Here are some problems faced by many traditional bike rentals organizations which are as follows:

* The country lacks a proper and comprehensive bike rental system, despite the presence of 3000 bike rental systems worldwide.
* Existing bike rental stations encounter difficulties with liability, paperwork, and time management.
* This absence of an effective bike rental system hinders convenient transportation options for individuals without access to their own vehicles.
* There is a need to address these challenges and establish a well-structured bike rental system to improve accessibility and streamline management processes.
* There is a need to improve the payment methods to ensure seamless and secure transactions, enhancing the overall rental experience.

Objectives

Here are some objectives of this system which are as follows:

* To develop a proper and comprehensive bike rental system for the country, considering the presence of 3000 bike rental systems worldwide.
* To address the challenges faced by existing bike rental stations, specifically related to liability, paperwork, and time management.
* To enhance transportation options by establishing an effective bike rental system for individuals without access to their own vehicles.
* To improve accessibility and streamline management processes in the bike rental system.
* To implement efficient payment methods that ensure seamless and secure transactions, thereby enhancing the overall rental experience.

Scope and Limitation

The Online Bike Rental System (OBRS) is specifically designed for small and medium-sized organizations with a single authority. It aims to streamline and enhance the bike rental process by providing a centralized platform for managing rental details and records. With security-driven features, the system ensures that only authorized personnel can access and modify the information. The OBRS also offers scalability, allowing the addition of new depots or rental locations as needed. Furthermore, it assists in determining the optimal economic order quantity, minimizing costs associated with ordering or manufacturing bikes. By implementing effective policies for location, layout, and materials handling equipment, the OBRS facilitates efficient store operations.

Despite the advantages of the OBRS, the existing paper-based record maintenance system poses several limitations. Manual record keeping leads to redundancy, making data susceptible to loss or inconsistency during modifications. The lack of a secure digital platform exposes records to unauthorized access and tampering. Additionally, the time-consuming and costly nature of the current system hampers information retrieval and storage. The manual handling of data also increases the likelihood of errors and inaccuracies. Moreover, the limited storage capacity for physical records poses challenges in maintaining data for an extended period. By transitioning to the OBRS, organizations can overcome these limitations and enjoy an automated and secure platform that optimizes bike rental management, improving overall efficiency and convenience.

Report Organization

In this project, I have developed an Online Bike Rental System. It is a dynamic System. It can be maintained and changed easily because it is based on the database. It contains web pages that are generated in real-time. These pages include web scripting code, such as PHP. It is fully secured from unauthorized access.

Chapter 1: The introduction put emphasis on the overview, Problem Statement, Objectives, Scope, and limitations of the project.

Chapter 2: Requirement and Feasibility Analysis is the important section such as Requirement Analysis and Feasibility study. Requirement Analysis explains the Functional and Non-Functional requirements of the project, and Feasibility Study explains why/how the project was practically implemented.

**Chapter 3:** System Design gives the design of the system developed so that it can be used during the project implementation.

**Chapter 4:** Implementation provides an indication of how the system is implemented, and what tools/platform has been used. Testing clarifies the system workflow.

**Chapter 5:** Conclusion marks on the end of the document by submitting the entire project and also opening the door further for research in improving the developed system. The lesson learned is also included in this chapter.

CHAPTER 2: BACKGROUND STUDY AND LITERATURE REVIEW

2.1 Background Study

Bike rental systems have seen significant growth and popularity in recent times, with nearly 3000 such systems operating worldwide. In the past, bike rental operations were managed manually, requiring customers to visit physical rental stations to rent bikes. However, the advent of the internet has transformed the bike rental landscape, enabling users to search and book rental bikes online with ease. Renowned bike rental services like snapbikes, ONN bike rentals, and bykemaina have emerged, catering to the increasing demand for flexible and convenient transportation solutions.

In response to this growing trend, the development of the Online Bike Rental System seeks to streamline and revolutionize the bike rental process. The proposed system will be web-based, providing an accessible and user-friendly interface for bike renters. Unlike traditional systems, the Online Bike Rental System will be solely focused on serving bike renters and managed by an admin responsible for overseeing rental operations within the organization. The system will not involve direct bike owners, who will instead act as renters when accessing bikes through the platform.

By implementing this web-based application, bike renters will experience enhanced convenience and accessibility in renting bikes. They can register, login, and browse through the available bike inventory, choosing their desired rental duration and making reservations online. On the other hand, the admin will be equipped with tools to efficiently manage the bike inventory, ensuring real-time updates and availability for renters.

The Online Bike Rental System aims to eliminate the need for physical visits to rental stations, providing a seamless and efficient online experience for users. The system's capabilities will extend beyond bike rental functionality, as it will generate comprehensive reports and transaction details, allowing the admin to monitor and optimize rental operations within the organization.

2.2 Literature Review

Mdnfmd ndnfm nmd

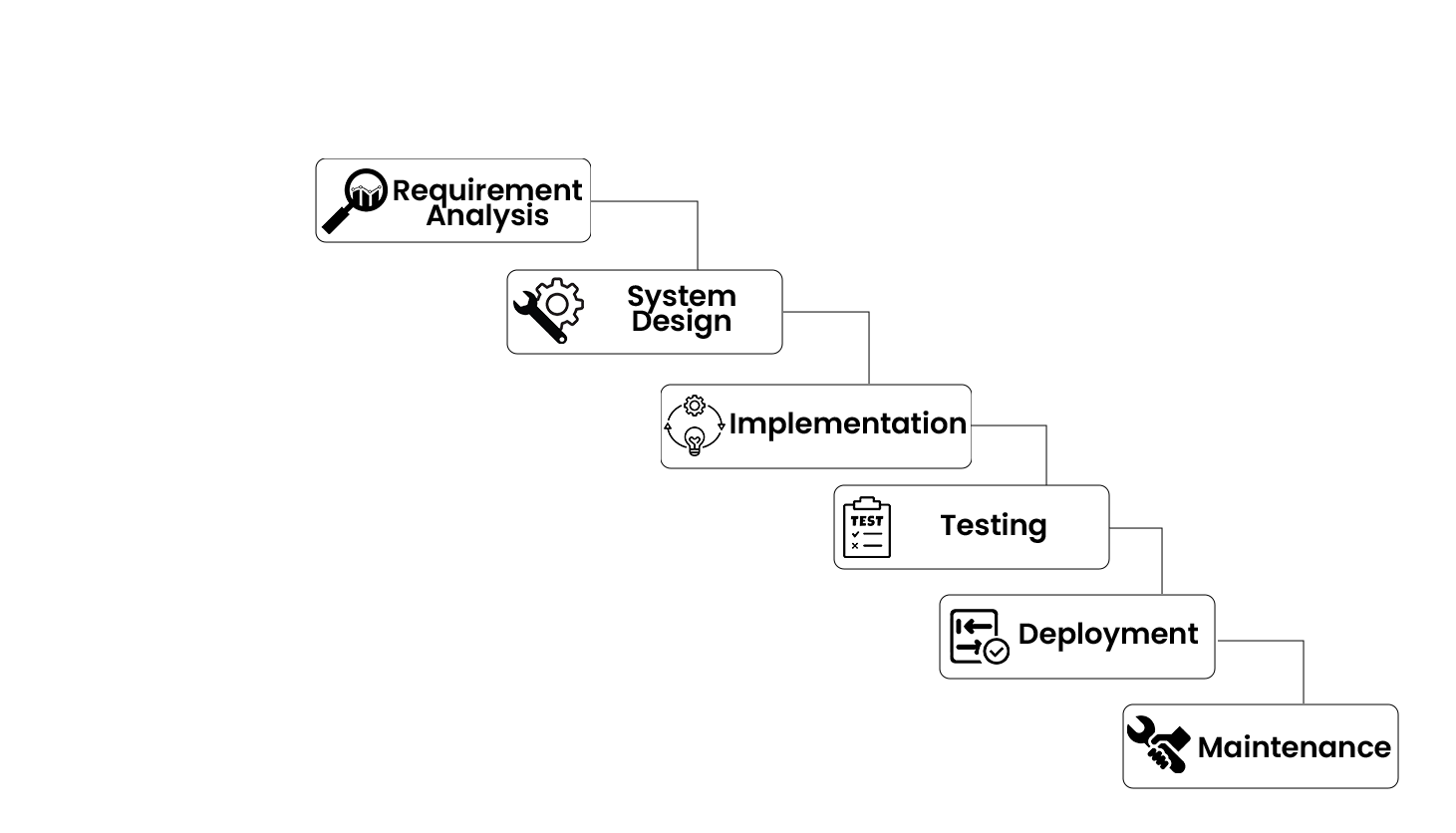
CHAPTER 3: SYSTEM ANALYSIS AND DESIGN

3.1 System Analysis

This system is commenced using the protocols of the waterfall model and All the tasks are started after completing the previous task. After setting the aims and objectives of the project, I have achieved the requirement and goals of the project. I followed both qualitative and quantitative analysis.

For this project, I followed the Waterfall model. The waterfall model is a linear, sequential approach to the software development life cycle (SDLC) that is popular in software engineering and product development. The waterfall model emphasizes a logical progression of steps. Waterfall models give the luxury to change the design or system requirements. It allows any alteration or modification to the plan as any coding hasn’t taken place. The Waterfall web development approach suits best for small projects, especially the ones under milestones and data-focused patterns. The life cycle of small projects completes in a short time that can be revised quickly for early launch and deployment.

This project Online Bike Rental System (OBRS) requires the development of various modules and for them to work with each other in conjunction, proper planning ahead is required. The waterfall modal is chosen for my project since it has a fixed structure and progress path for each of the design and development steps.



3.1.1 Requirement Analysis

Requirement identification is the first phase of an SDLC where requirements and specifications of the software are collected. For gathering requirements, we have studied the existing system. We have performed a literary analysis of the existing systems as well as reviewed problems and lack of features on systems to include in this project. There there two types of Requirement on the basis of the analysis.

1. Functional Requirement

The functional requirements are things that the system must do for a user. Here are my some of functional requirement according to my system in renter and admin module.

|  |  |
| --- | --- |
| **Admin Module** | **Renter Module** |
| * Login * Accepts Bike request * Manage Users * Manages Rental Bikes * View Rental Report of Each month * Approve me the rental request. * Set the prices of bike for rent. * Update the information | * Signup/Login * Request for bike * Contact Admin * Pay money through online. * View Rented bikes. * Filter out the bike information |

1. Non-functional Requirement

It describes system elements that are concerned with how the system fulfills functional requirements. They are as follows:

* **Security**: The system should be secure from external entities.
* **Performance and Response Time**: The system should perform in an efficient way for the client.
* **Error Handling**: There must be less error. If any such error is detected, an error message should be supplied to help the user through the recovery process
* **Availability**: The system should be easily available to the user.
* **Ease of Use**: The system should be user-friendly.

3.1.2 Feasibility Analysis

It is an analysis of a proposed project to determine whether it is feasible and should go ahead Including economic, technical, operational, and scheduling considerations

* **Operational Feasibility**

The system is operationally practical since it can be used by ordinary users with basic computer abilities who do not require any further training. We created this system with the willingness and capacity to design, administer, and run a system that is simple for end-users to use.

* **Technical Feasibility**

A large part of determining resources has to do with assessing technical feasibility. It considers the technical requirements of the proposed project. The technical requirements are then compared to the technical capability of the organization. The systems project is considered technically feasible if the internal technical capability is sufficient to support the project requirements.

* **Economic Feasibility**

The economic analysis could also be referred to as cost/benefit analysis. It is the most frequently used method for evaluating the effectiveness of a new system. In economic analysis, the procedure is to determine the benefits and savings that are expected from a candidate system and compare them with costs. If the benefits outweigh costs, then the decision is made to design and implement the system. An entrepreneur must accurately weigh the cost versus benefits before taking action.

* **Schedule Feasibility**

Analysis is taken, how much time is taken to compete this project. Studied about what are the factors are taking much time. Find the ways to deliver the system as soon as possible. This assessment is the most important for project success; after all, a project will fail if not completed on time. In scheduling feasibility, an organization estimates how much time the project will take to complete.

3.2 System Design

Software architecture is the blueprint of building software. It shows the overall structure of the software, the collection of components in it, and how they interact with one another while hiding the implementation.

3.2.1 Data Modeling(ER-Diagram)

An Entity-Relationship diagram shows the relationship among entity sets. An entity set is a group of similar entities and these entities can have attributes.

3.2.2 Process Modeling (DFD diagram)

3.2.3 Use Case Diagram

3.2.4 Database Schema Design

3.2.5 Interface Design

3.2.6 Sequence Diagram

3.3 Algorithm Details

Here are the some Algorithm which I had used in my project.

1. Breaking array into sub arrays

Step-by-step algorithm for function convert\_subarrays(sublength):

**Step 1:** Initialize variables:

- Set k = 0

- Create an empty array sub\_dates\_array

- Create an empty array sub\_bikecounts\_array

**Step 2:** Outer loop:

- For i from 0 to Math.floor(dates.length / sublength) - 1, do the following:

**Step 2.1:** Create subarrays for dates and bikecounts:

- Create an empty array for sub\_dates\_array[i]

- Create an empty array for sub\_bikecounts\_array[i]

**Step 2.2:** Inner loop:

- For j from 0 to sublength - 1, do the following:

- If k is greater than or equal to dates.length, exit the inner loop.

**Step 2.2.1:** Assign values to subarrays:

- Assign dates[k] to sub\_dates\_array[i][j]

- Assign bikecounts[k] to sub\_bikecounts\_array[i][j]

**Step 2.2.2:** Increment k by 1

**Step 3:** Return the result:

- Return a dictionary containing two arrays:

- 'Dates' with the sub\_dates\_array

- 'bikecounts' with the sub\_bikecounts\_array

End of function

CHAPTER 4: IMPLEMENTATION AND TESTING

4.1 Implementation

Implementation of a system is as much important as the creation of it. Implementation can easily destroy the good work done in the earlier phases and bring the system to a standstill. Implementation requires technical and managerial skills as the implementers work as change agents.

4.1.1 Tools Used

Frontend tools

1. Html
2. CSS
3. Javascript
4. Chartjs

Backend tools

1. Laravel
2. Mysql

Server

1. Xammp Server

Code Editor

1. Visual Code Editor

Version Control

1. Git and GitHub

Online Payment Integration

1. Khalti

4.2 Testing

Testing is an integral part of the software development process. It is the process of verifying and validating that software meets the requirements based on the design and development proposed. This Project is focused on the validation and verification of the user input data from the very beginning. A single module is created for the application part of the project. The Project is broken down into several modules and configured as necessary per requirements. Testing is performed after completion of each module and after their integration also.