



**Tribhuvan University**

**Faculty of Humanities and Social Sciences**

**A Project Report**

**On**

**“Online Bike Rental System”**

**Submitted to:**

**Department of Computer Application**

**Lumbini ICT College**

**Gaindakot, Nawalpur, Nepal**

*In partial fulfillment of the requirements  
for the Bachelors in Computer Application*

**Submitted by:**

**Bipin Paudel**

**TU Roll No: 6-2-1194-68-2019**

**Under the Supervision of**

**Department of Computer Application**

**August 2023**



**Tribhuvan University**

**Faculty of Humanities and Social Sciences**

**Lumbini ICT Campus**

### **Supervisor's Recommendation**

I hereby recommend that this project prepared under my supervision by **Bipin Paudel** entitled “**Online Bike Rental System**” in partial fulfillment of the requirements for the degree of Bachelor of Computer Application is recommended for the final evaluation.

.....

**Er. Mod Nath Acharya**

Supervisor /Lecturer/Co-ordinator

Department of Computer Application

Lumbini ICT Campus

Gaindakot, Nawalpur, Nepal

## **Abstract**

Online Bike Rental System, crafted using Laravel, with a focus on two primary user roles - Admin and Renter. This system empowers renters to effortlessly list their bikes for rent, facilitating seamless transactions for potential renters. The Admin holds the responsibility of managing all administrative data, ensuring smooth operations throughout the platform. Additionally, the Admin gains access to comprehensive monthly rental reports, enhancing decision-making capabilities. The convenience of online payment options for renters further enhances the user experience, simplifying the rental process. Notably, the system incorporates a Rental Management feature, where bike rentals necessitate prior approval from the Admin before they become available for rent, ensuring a secure and reliable rental environment. Online Bike Rental System aims to create a user-friendly, secure, and transparent platform that streamlines bike rental transactions, providing a beneficial experience for bike owner companies and renters alike.

## **Acknowledgement**

Firstly, I would like to express our deep gratitude to the Department of Bachelor in Computer Applications (BCA), Lumbini ICT College for providing the necessary resource and guideline to come up with this project. We would like to thank Head of Department and all the concerned faculty member of the department. Our special thanks goes to our project teacher Sudip Parajuli and project supervisor Er.Mod Acharya sir for his insightful advice, motivating suggestions, invaluable guidance, help and support in successful completion of this project and also for his/her constant encouragement and advice throughout our Project Development. We express our deep gratitude to Er. Mod Acharya, for their regular support, co-operation, and co-ordination. Special thanks also goes to all the teaching and non-teaching staff as well as colleagues, who have been assisting in the project directly as well as indirectly by reviewing and suggesting the improvements. Last but not the least, our cordial thank goes to Lumbini ICT College Administration who motivated us to go ahead with this project.

## Table of Contents

<b>Abstract.....</b>	<b>ii</b>
<b>Acknowledgement .....</b>	<b>iii</b>
<b>Table of Contents .....</b>	<b>iv</b>
<b>List of Figures.....</b>	<b>v</b>
<b>List of Tables .....</b>	<b>vi</b>
<b>CHAPTER 1: INTRODUCTION.....</b>	<b>1</b>
1.1 Introduction.....	1
1.2 Statement of Problem.....	1
1.3 Objectives .....	2
1.4 Scope and Limitation .....	2
1.5 Report Organization.....	3
<b>CHAPTER 2: BACKGROUND STUDY AND LITERATURE REVIEW .....</b>	<b>4</b>
2.1 Background Study.....	4
2.2 Literature Review.....	4
<b>CHAPTER 3: SYSTEM ANALYSIS AND DESIGN .....</b>	<b>5</b>
3.1 System Analysis.....	5
3.2 System Design .....	9
3.2 Database Design.....	14
3.3 Algorithms .....	15
<b>CHAPTER 4: IMPLEMENTATION AND TESTING .....</b>	<b>16</b>
4.1 Implementation .....	16
4.2 Testing.....	19
<b>CHAPTER 5: CONCLUSION AND FUTURE RECOMMENDATIONS.....</b>	<b>26</b>
5.1 Lesson learnt/outcomes.....	26
5.2 Conclusion .....	26
5.3 Future Recommendations .....	27
<b>Appendices.....</b>	<b>28</b>
Logsheet.....	28
Screenshots .....	29
<b>References .....</b>	<b>32</b>

## List of Figures

Figure 1: Waterfall model .....	5
Figure 2: Use-Case Diagram.....	9
Figure 3: Rent Bike Activity Diagram .....	10
Figure 4: Add Bike Activity Diagram .....	11
Figure 5: Class Diagram .....	12
Figure 6: Sequence Diagram.....	13
Figure 7: ER-Diagram.....	14
Figure 8: Database Schema.....	14
Figure 9: Logsheet .....	28
Figure 10: Bikes Table .....	29
Figure 11: Admin Dashboard.....	29
Figure 12: Rents Table .....	30
Figure 13: Analytics.....	30
Figure 14: Show Bikes.....	31

## **List of Tables**

Table 1: User Modules .....	6
Table 2: Time Plan .....	8
Table 3: Sign in Test Case .....	19
Table 4: Sign Up Test Case .....	20
Table 5: Test Case For Renter Module.....	21
Table 6: Test Case For Admin Module .....	22
Table 7: Test Cases For System testing.....	25

# **CHAPTER 1: INTRODUCTION**

## **1.1 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.

## **1.2 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.



### **1.3 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.

### **1.4 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.

## 1.5 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 become increasingly popular worldwide, with nearly 3000 such systems in operation. In the past, bike rentals were managed manually, requiring customers to visit rental stations in person. But with the internet, the landscape has changed. Renowned services like snapbikes, ONN bike rentals, and bykemainia now offer online booking for rental bikes, meeting the demand for flexible and convenient transportation solutions.

To further enhance this trend, we propose an Online Bike Rental System that will revolutionize the rental process. This web-based system will provide a user-friendly interface for renters, allowing them to easily browse and book bikes online. An admin will manage the system, overseeing rental operations. Unlike traditional systems, bike owners will act as renters when accessing bikes through the platform. The web-based application will bring enhanced convenience to bike renters, who can register, log in, and reserve bikes online, selecting their preferred rental duration. The admin will efficiently manage the bike inventory, ensuring real-time updates and availability for renters [1].

By eliminating the need for physical visits to rental stations, the Online Bike Rental System aims to provide a seamless online experience. Additionally, the system will generate comprehensive reports and transaction details, empowering the admin to monitor and optimize rental operations efficiently.

## **2.2 Literature Review**

Online Bike Rental System has been widely used in order to provide an instant bike confirmation service to the website's visitors, hopefully turning them into bookers, renters, etc. An Online Bike Rental System allows people to receive reservations with thin twenty-four hours a day. A lot of Bike Rental companies are now using the power of the internet to make their business successful because just like what it is easier to manage in computer or in the database rather than the file or self as well as the ability to place or reversation [2].

## CHAPTER 3: SYSTEM ANALYSIS AND DESIGN

### 3.1 System Analysis

The Online Bike Rental System (OBRS) project was initiated following the principles of the Waterfall model, which is a linear and sequential approach to the software development life cycle (SDLC). This model emphasizes completing each task before moving on to the next, ensuring a logical progression of steps. After establishing the project's aims and objectives, I successfully achieved the required goals by employing both qualitative and quantitative analysis.

The Waterfall model proved suitable for this project due to its fixed structure and progress path for design and development. With a clear plan in place, it allowed for effective coordination between various modules, enabling them to work together seamlessly. This approach is particularly well-suited for small projects, especially those with milestones and data-focused patterns, as it facilitates a swift and streamlined life cycle, perfect for quick revisions, early launch, and smooth deployment. By following the Waterfall model [3], I ensured that the Online Bike Rental System was developed efficiently, meeting the specified requirements and objectives.

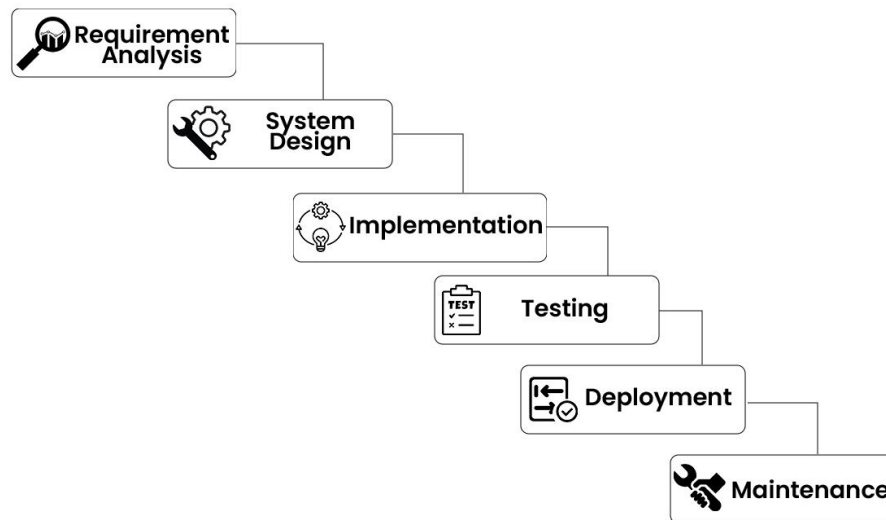


Figure 1: Waterfall model

### 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 [4].

#### i. 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.

*Table 1: User Modules*

Admin Module	Renter Module
<ul style="list-style-type: none"><li>• Login</li><li>• Accepts Bike request</li><li>• Manage Users</li><li>• Manages Rental Bikes</li><li>• View Rental Report of Each month</li><li>• Approve me the rental request.</li><li>• Set the prices of bike for rent.</li><li>• Update the information</li></ul>	<ul style="list-style-type: none"><li>• Signup/Login</li><li>• Request for bike</li><li>• Contact Admin</li><li>• Pay money through online.</li><li>• View Rented bikes.</li><li>• Filter out the bike information</li></ul>

#### ii. 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.
- **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 complete 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.

**Table 2: Time Plan**

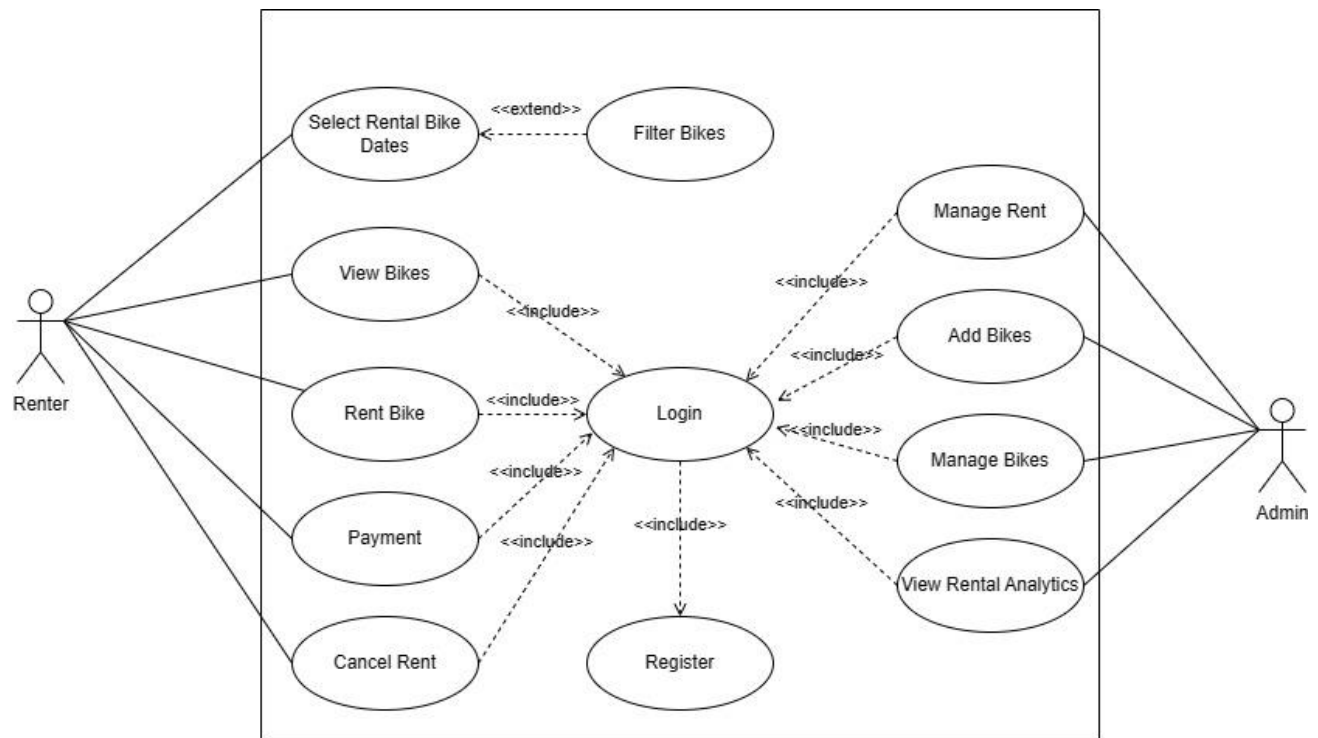
<b>Weeks</b> ➡  <b>Activities</b> ⬇	1	2	3	4	5	6	7	8	9	10	11	12	13
<b>System Analysis</b>	1 W												
<b>System Design</b>		2W											
<b>Coding &amp; Implementation</b>				6W									
<b>Testing</b>										2W			
<b>Documentation</b>	13W												

## 3.2 System Design

Software design is the blueprint of building system. It shows the overall structure of the system, the collection of components in it, and how they interact with one another while hiding the implementation [5].

### a) Context Modeling

Use-case diagrams illustrate and define the context and requirements of either an entire system or the important parts of the system.



**Figure 2: Use-Case Diagram**



## b) Process Modeling

We use Activity Diagrams to illustrate the flow of control in a system and refer to the steps involved in the execution of a use case.

### Rent Bike – Usecase – Activity Diagram

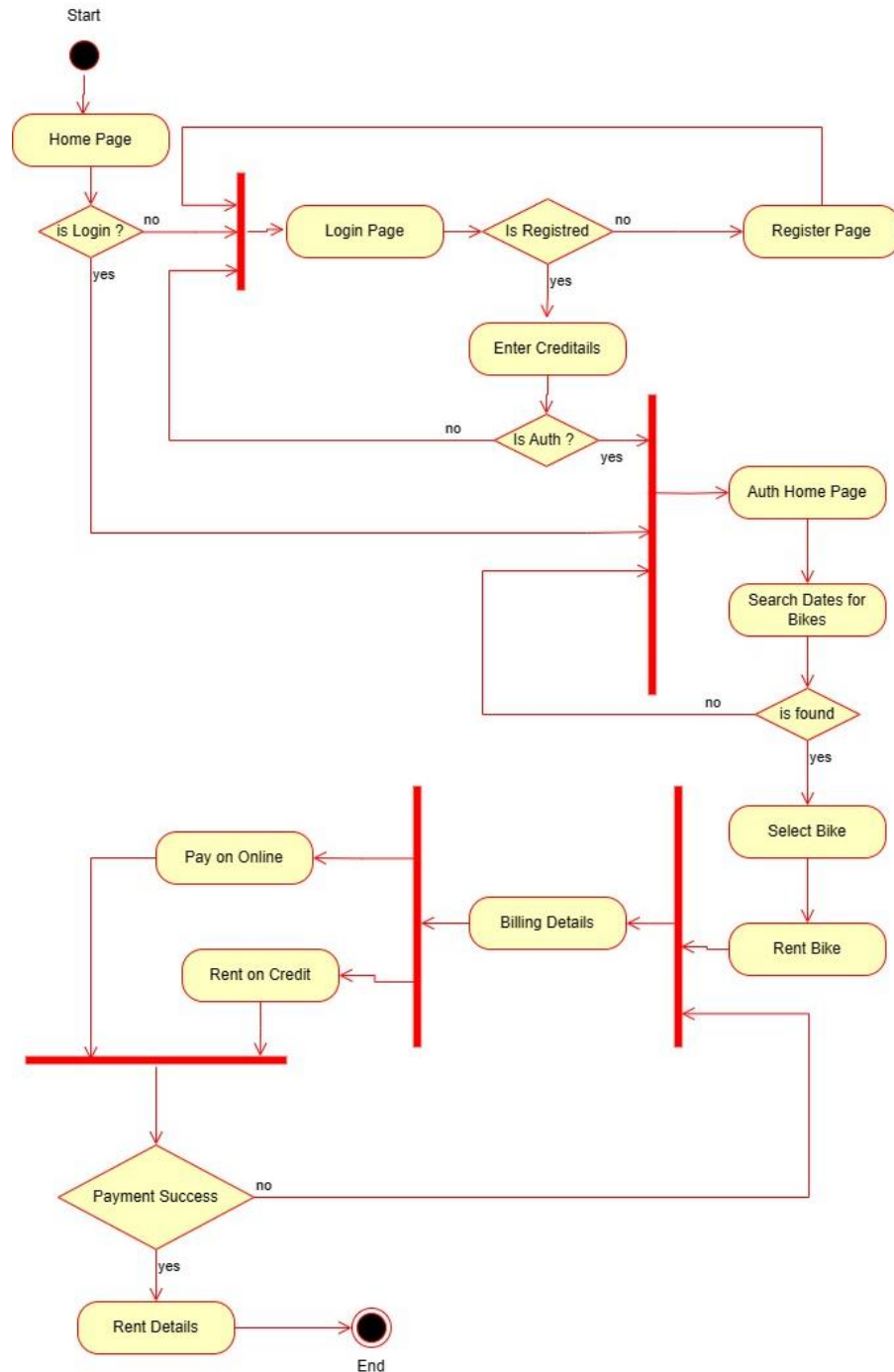
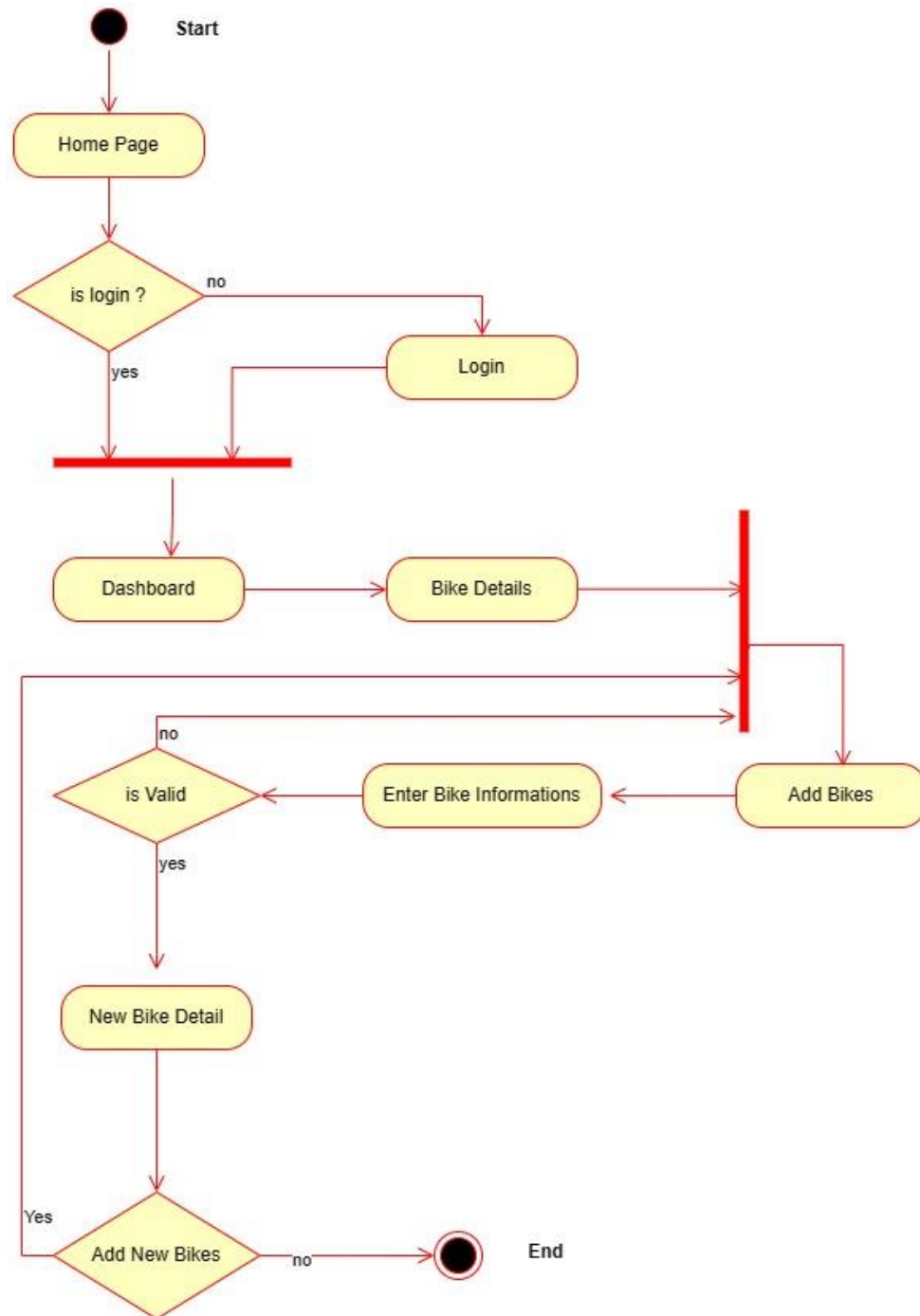


Figure 3: Rent Bike Activity Diagram

- Add Bike – Usecase- Activity Diagram



**Figure 4: Add Bike Activity Diagram**

### c) Structural Modeling

Structural modeling captures the static features of a system. Here I had used class diagram.

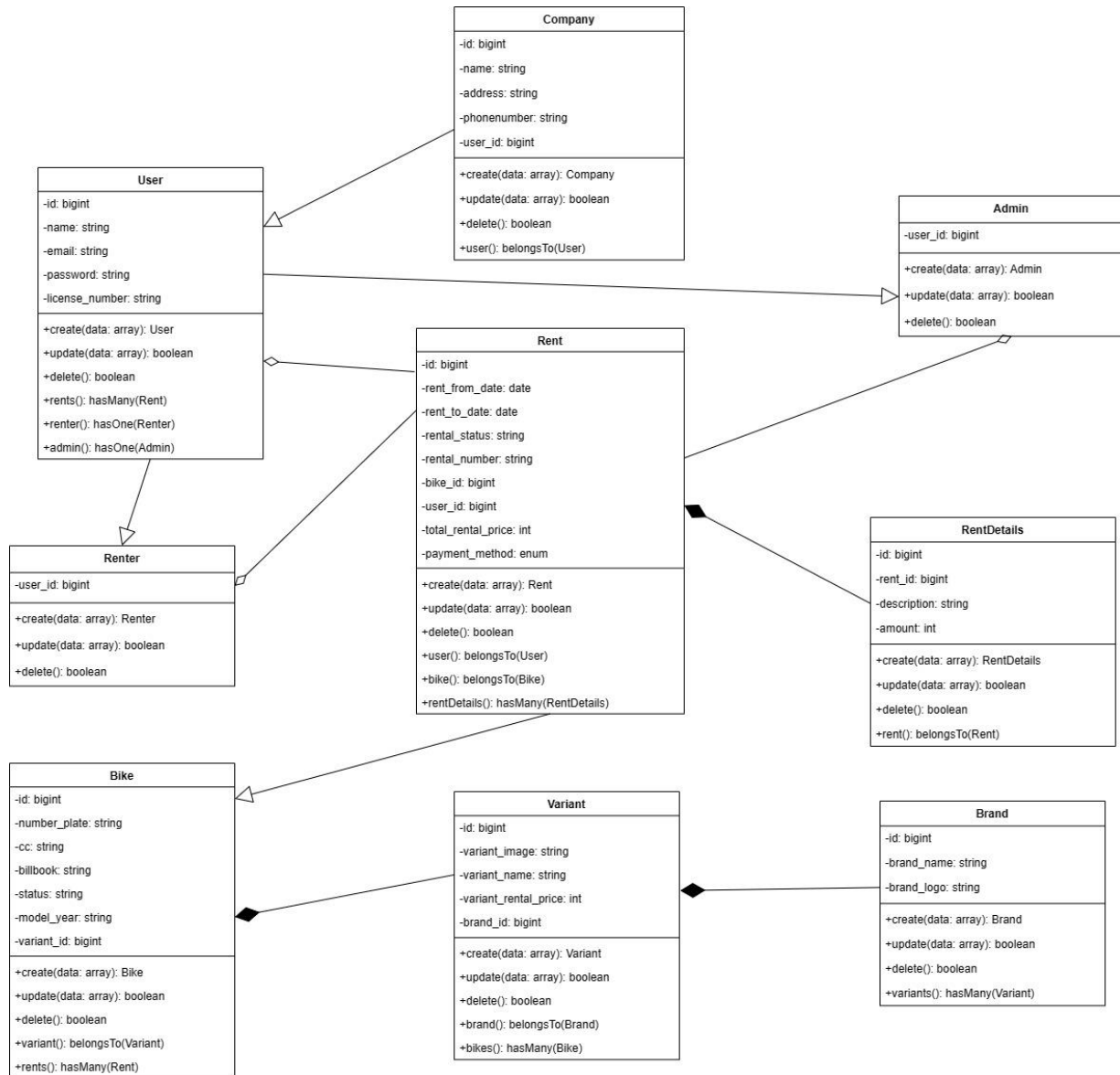


Figure 5: Class Diagram

#### d) Dynamic Modeling

The dynamic model is used to express and model the behavior of the system over time. Here I had used Sequence Diagram to show Dynamic modeling.

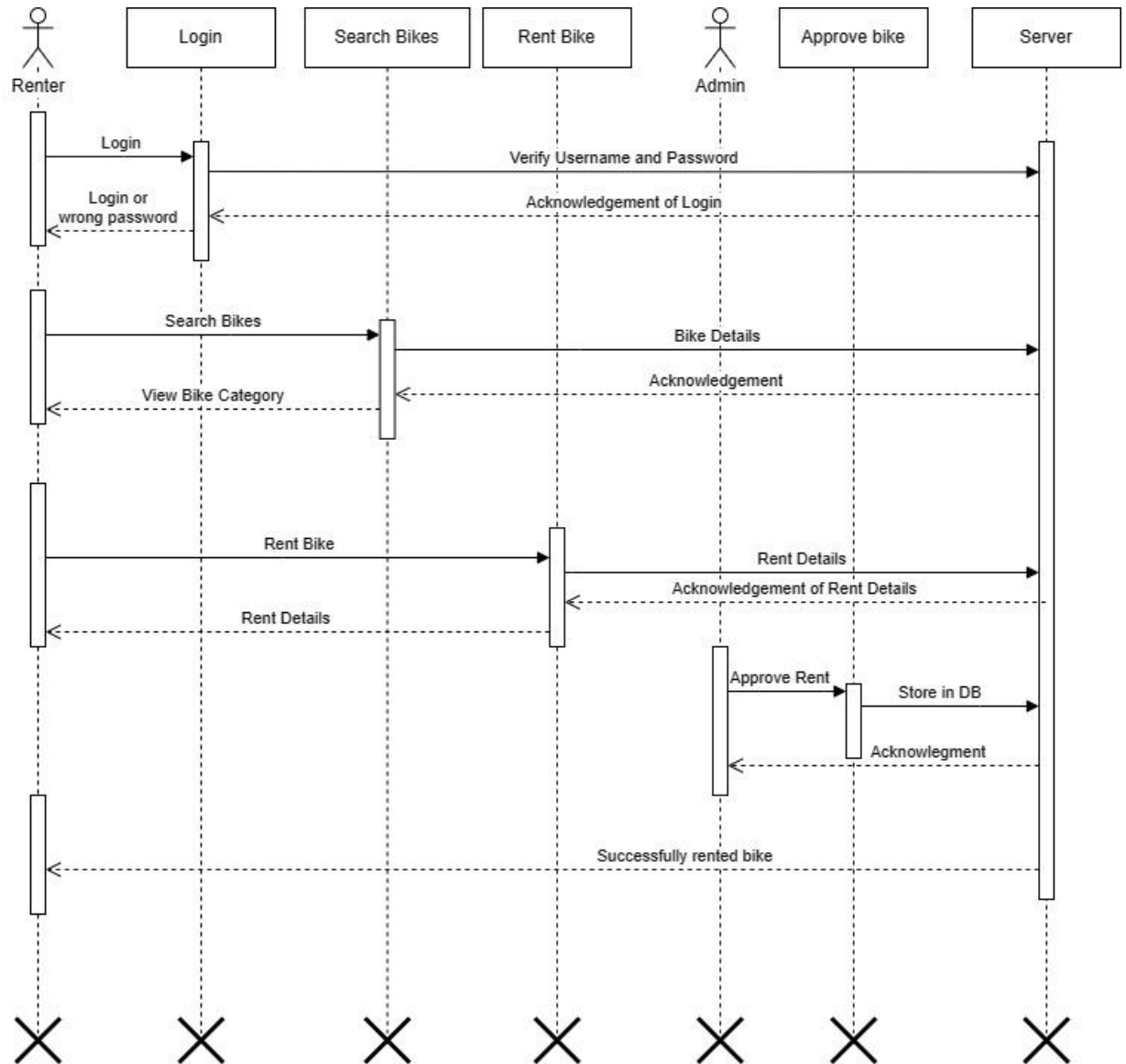


Figure 6: Sequence Diagram

## 3.2 Database Design

Database design is the organization of data according to a database model.

### a) ER Diagram

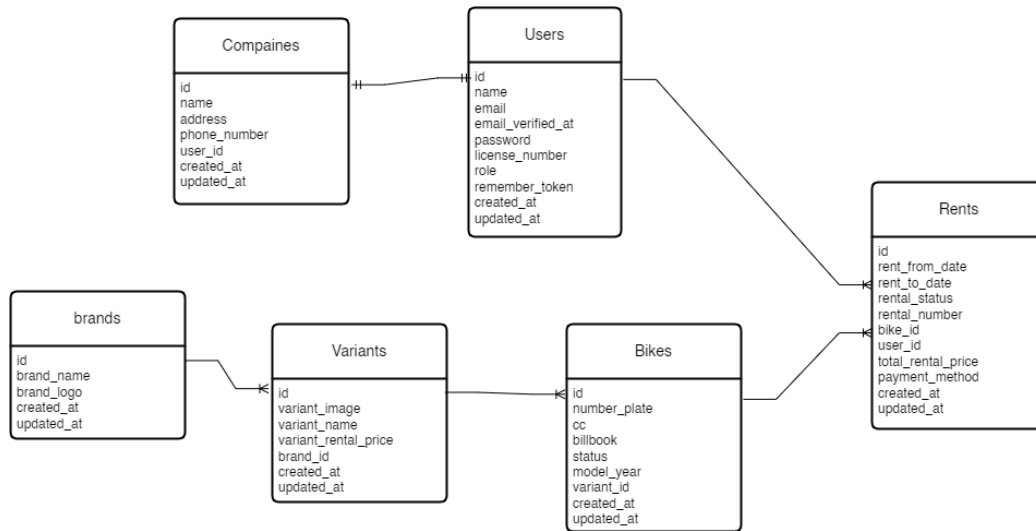


Figure 7: ER-Diagram

### b) Database Schema

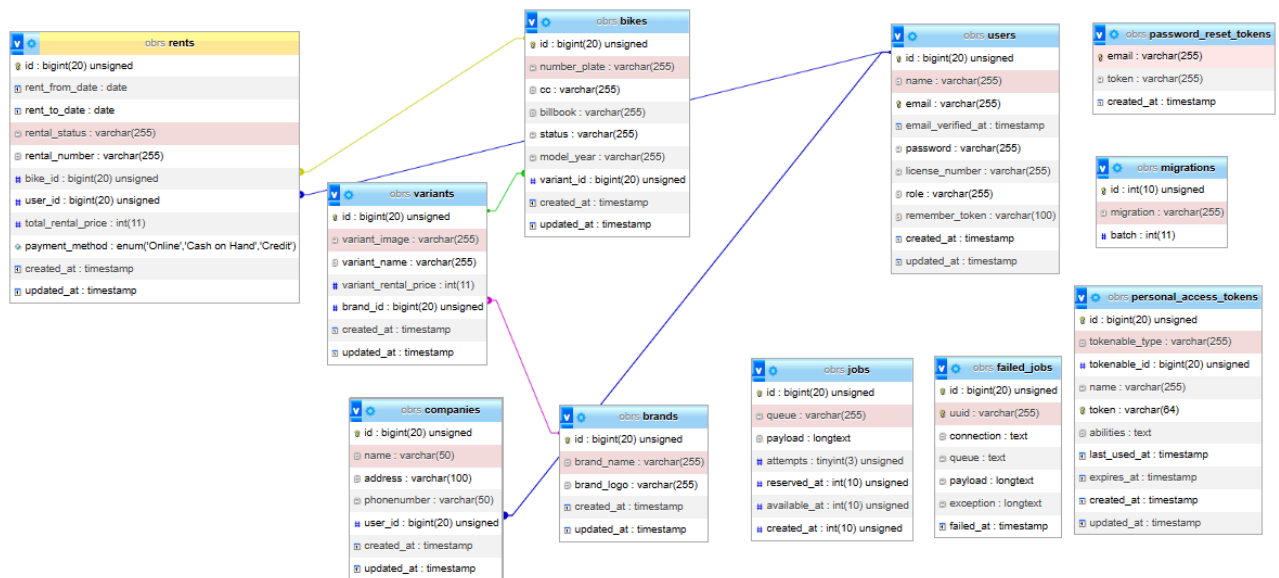


Figure 8: Database Schema

### 3.3 Algorithms

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

#### I. Breaking array into sub arrays:

**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  $\text{Math.floor}(\text{dates.length} / \text{sublength}) - 1$ , 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  $\text{sublength} - 1$ , 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` and '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**

##### **i. Html**

HTML is the standard markup language used to create the structure and content of web pages. It provides a set of elements (tags) that define the different parts of a webpage, such as headings, paragraphs, images, links, forms, and more. HTML is the backbone of any web page and is responsible for defining the structure of the content.

##### **ii. CSS**

CSS is a style sheet language used to control the presentation and layout of HTML documents. It enables web developers to define the colors, fonts, spacing, and overall appearance of a webpage. By separating the content (HTML) from the presentation (CSS), developers can create visually appealing and consistent designs across multiple pages.

##### **iii. Javascript**

JavaScript is a high-level, interpreted programming language that allows developers to add interactivity and dynamic elements to web pages. With JavaScript, you can perform actions such as form validation, image sliders, interactive maps, and more. It plays a crucial role in enhancing user experience and making websites more interactive.

##### **iv. Chartjs**

Chart.js is a popular JavaScript library that simplifies the process of creating various types of interactive and visually appealing charts and graphs on web pages. It supports bar charts, line charts, pie charts, and more, providing a user-friendly API to render data in a graphical format [6].

## **Backend tools**

### **i. Laravel**

Laravel is a PHP-based open-source web application framework that simplifies the development process by providing a robust and elegant syntax. It follows the Model-View-Controller (MVC) architectural pattern, making it easier to build scalable and maintainable web applications. Laravel offers various features like routing, database migration, ORM (Object-Relational Mapping), and templating engine, making it a popular choice among web developers .

### **ii. Mysql**

MySQL is a widely used open-source relational database management system (RDBMS). It is often chosen as the backend database for web applications due to its performance, scalability, and ease of use. MySQL stores and manages data, making it accessible for the web application to retrieve and manipulate as needed.

## **Server**

### **i. Xampp Server**

XAMPP is a cross-platform web server solution that provides the necessary tools to set up a local development environment. It includes Apache (web server), MySQL (database), PHP, and Perl, making it easy to create and test web applications on a personal computer before deploying them to a live server. XAMPP simplifies the process of configuring and managing a local server environment for web development.

## **Code Editor**

### **i. Visual Code Editor**

Visual Studio Code is a popular, free, and open-source code editor developed by Microsoft. It offers a highly extensible and customizable environment for web development and supports a wide range of programming languages. VS Code provides features like syntax highlighting, code completion, debugging, version control integration, and a vast collection of extensions that enhance productivity for developers.



## **Version Control**

### **i. Git**

Git is a distributed version control system that allows developers to track changes to their codebase over time. It enables collaborative development, easy branching, and merging of code, and provides a history of all modifications made to the project. Git is widely used in software development to manage and coordinate code changes among team members efficiently.

## **Online Payment Integration**

### **i. Khalti**

Git is a distributed version control system that allows developers to track changes to their codebase over time. It enables collaborative development, easy branching, and merging of code, and provides a history of all modifications made to the project. Git is widely used in software development to manage and coordinate code changes among team members efficiently.

## 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 [7].

### 4.2.1 Test Cases For Unit Testing

Unit Testing was done to test field validations, navigation, and functionality of the programs and its blocks. These tests are applied on various functions within each program and other critical program blocks. Table 1, 2 and 3 outline two sample test cases for Unit Testing performed on the system.

#### I. Test case for Sign in

*Table 3: Sign in Test Case*

Test Serial	Test Case	Scenario	Step	Test Data	Expected Result
1	Sign in	User enters wrong email	Go to the login page and enter email and password	Email: <a href="mailto:xyz@gmail.com">xyz@gmail.com</a> Password: nepal@123	No email found
2	Sign in	User enters the wrong password	Go to login page and enter email and password.	Email: <a href="mailto:obrs@gmail.com">obrs@gmail.com</a> Password: abcd@123	Invalid Password.

3	Sign in	User enters the correct email and password	Go to login page and email and password	Email : <a href="mailto:xyz@gmail.com">xyz@gmail.com</a> Password: nepal@123	Successfully Sign in and redirected to Home page
---	---------	--	---	---	--

## II. Test Case for Sign up

*Table 4: Sign Up Test Case*

Test Serial	Test Case	Scenario	Step	Test Data	Expected Result
1	Sign Up	User Forget to enter a particular field	Go the the Sign Up Page and fill the required informations.	Name: Rohan Sharma Password: Nepal@123 Confirm Password: Date of Birth: Email: License Number :	Please fill the all the fields !
2	Sign Up	User enters the Already registered email	Go to the Sign Up page and enter the already registered email	Email: <a href="mailto:obrs@gmail.com">obrs@gmail.com</a>	This email is already registered.
3	Sign Up	User enters the different password	Go to Sign up page and enter	Password: nepal@123	Password and confirm

		and Confirm password.	different password and confirm password	Confirm Password: india@123	password do not match!
--	--	--------------------------	--	-----------------------------------	---------------------------

### III. Test Case for Renter module

*Table 5: Test Case For Renter Module*

Test Serial	Test Case	Scenario	Step	Test Data	Expected Result
1	Find Bikes	To Find Bikes.	Go to home page and select the rent from date and rent to date and search the bike.	From : 2023/04/02 To 2023/04/12`	Bikes in the date are shown in bike catalog.
2	Rent a Bike	To rent a bike.	Select a bike and see the details click rent bike and select the payment method credit or online.	Payment method: Khalti	Successfully rented a bike.

3	View Rented history	User clicked on rent on bikes	Click on user menu in navbar and click rented bike history	Rented bike history	Rented Bike history is shown.
4	Cancel Rental Booking	To Cancel Rental Booking	Click on cancel rental booking button on Rental details	Cancel Rental Booking	Rent is canceled Successfully.

#### IV. Test Case for Admin

**Table 6: Test Case For Admin Module**

Test Serial	Test Case	Scenario	Step	Test Data	Expected Result
1	View Brands	Admin wants to see brands	Click on brands on side bar.	Brands	Brands table is shown in /brands
2	Add Brand	Admin wants to add new brand	Click on add brand button	Brand name: Yamaha	New Brand Added successfully.
3	View Variants	Admin wants to see variants.	Click on variants on side bar	Variants	Variants table is shown in /variants.

4	Add Variants	Admin wants to add new variants	Click on add variant button	Variant name: FZ Cc: 180 Image: variant.jpg Brand: Yamaha Price: 220	New Variant Added successfully
5	View Bikes	Admin wants to see Bikes.	Click on Bikes on sidebar	Bikes	Bike table is shown in /variants.
6	Add Bikes	Admin wants to add new Bikes	Click on add bike button	Bike plate number: 6348 Cc: 180 Bill book copy: billbook.jpg variant: FZ Status: Available	New Bike Added successfully
7	View Rents	Admin wants to see Rents.	Click on rents on sidebar.	Rents	Rents table is shown in /rents.
8	Add rent	Admin wants to add rent for the user	Click on add rent and fill the all the informations.	Brand: Yamaha Variant: FZ Bike:3434 From date: 2023/05/23 To Date: 2023/05/27 Renter email: ram@gmail.com	New Rent Added successfully

9	Change Report month	Admin wants to view report and change month	Click on date and change month	Month: July	Month Changed Successfully
---	---------------------------	---	--------------------------------------	-------------	----------------------------------

#### 4.2.2 Test Cases For System Testing

The focus of the system testing is to evaluate the compliance of the entire system with respect to the specified requirements. System testing helps in approving and checking the business, functional, technical, and any non-functional requirements of the application concerning the architecture as a whole.

**Table 7: Test Cases For System testing**

<b>Sn</b>	<b>Test Case</b>	<b>Excepted Result</b>	<b>Actual Result</b>	<b>Remarks</b>
1	User Registration	Register User	Register User	Success
2	User Login	Login User	Login User	Success
3	Adding Bikes	Bike Added	Bike Added	Success
4	Updated bike details	Bike details updated	Bike details updated	Success
5	Deleting bike	Bike deleted	Bike deleted	Success
6	Adding Variant	Variant Added	Variant Added	Success
7	Updating Variant	Variant Updated	Variant Updated	Success
8	Deleting Variant	Variant Deleted	Variant Deleted	Success
9	View Rent	Rent Shown	Rent Shown	Success
10	Approve Rent	Rent Approved	Rent Approved	Success
11	Cancel Rent	Rent Cancel	Rent Cancel	Success
12	Rent Status Change	Rent Status Change	Rent Status Change	Success
13	Send mail	Send mail	Send mail	Success
14	Logout	Logout	Logout	Success



## **CHAPTER 5: CONCLUSION AND FUTURE RECOMMENDATIONS**

### **5.1 Lesson learnt/outcomes**

Through the project from Lumbini ICT, the lesson learnt were:

- Time Management.
- Handling of urgent bugs and fixing them immediately.
- Making planning before starting task.
- Brainstorming and playing with ideas.
- Researching the particular content.
- Implementing real world practical actions into coding functions.
- Working as a software developer can be considered as a potential biker.

It has helped in gaining knowledge about various technical tools and frameworks used in software development and the process that should be followed for the proper development completion. It has developed researching and professional skills in me.

### **5.2 Conclusion**

An online bike rental system is a web-based application that allows a person to rent/reserve a bike. This bike is successfully implemented with all the features mentioned in system requirement specification. Traditionally, people have to wait for their turn so it is time consuming process which consumes more time to do manual paperwork. Bike Rental System is successfully implemented using HTML, CSS, JavaScript, Tailwind, Laravel and MySQL which is open source and freely available on the internet and it successfully solves the problem of traditional renting system. The proposed system is useful for people with minimal IT knowledge with the use of internet.

After the completion of the system users will be able to use an online system for bike renting of their desired choice. Bike Renter can create an account to rent the bike and bike owner create account and rent a bike and register his bike on rent. Admin can see what activities are happening in the system and approve the rental bikes. One individual can also register their bike to give on rent.

### **5.3 Future Recommendations**

The success of this project will depend on the number of users who gets benefits from the system.

Future updates will be done after the review and feedbacks obtained from user using this system.

Future recommendation can be like:

- Adding forget password option.
- Download payment statement option.
- Bike rental rating and reviews.
- Better UI and UX.
- Rental date Seclude Calendar.

## Appendices

### Logsheets

**LUMBINI ICT CAMPUS**  
Gaundakot, Nawalpur  
Project Logbook

Project Title: *Online Bike Rental System.*  
Faculty: *BCA*  
Group Name:  
Student Name: *Bipin Paudel.*

Semester: *Sixth*  
Year: *2079/80*  
Supervisor Name:

Date	Description on the Functions/Features/Improvements	Supervisor's Signature	Remarks
<i>2080/1/12</i>	<ul style="list-style-type: none"> <li>- Requirement Elicitations and categorization, Requirement Verification.</li> <li>- System Modeling.</li> <li>- Database Modeling.</li> </ul>	<i>[Signature]</i>	
<i>2080/1/1</i>	<ul style="list-style-type: none"> <li>- Admin. Dashboard layout.</li> <li>- Brands, Variants &amp; Bike CRUD Operations.</li> <li>- Login and Register. Authentication.</li> </ul>	<i>[Signature]</i>	
<i>2080/3/01</i>	<ul style="list-style-type: none"> <li>- Admin Dashboard Completed.</li> <li>→ Analysis Part Completed.</li> <li>→ Rent bike process completed.</li> </ul>	<i>[Signature]</i>	

Figure 9: Logsheets

## Screenshots

The screenshot shows the 'Bikes' management page in the Poudel Bike Rentals application. On the left is a dark sidebar with navigation links: Dashboard, Users, Brands, Variants, Bikes, Rents, Analytics, and Company setup. The main content area is titled 'Bikes' and includes a table with columns: Sn, Bike Number Plate, cc, Bike Brand, Variant name, Bill Book, Model Year, Status, Added date, and Action. Above the table are controls for 'Show' (set to 10), 'Entries', 'Select Status' (set to 'Show all'), and an 'Add Bike' button. The table contains two entries: one with status 'Unavailable' and another with status 'Available'. Each entry has 'Edit' and 'Delete' buttons in the Action column.

Sn	Bike Number Plate	cc	Bike Brand	Variant name	Bill Book	Model Year	Status	Added date	Action
1	3233	130	Yamaha	FZ	<a href="#">View Bill Book</a>	2013	Unavailable	1 month ago	<a href="#">Edit</a> <a href="#">Delete</a>
2	23233	120	Yamaha	FZ	<a href="#">View Bill Book</a>	2019	Available	1 month ago	<a href="#">Edit</a> <a href="#">Delete</a>

Figure 10: Bikes Table

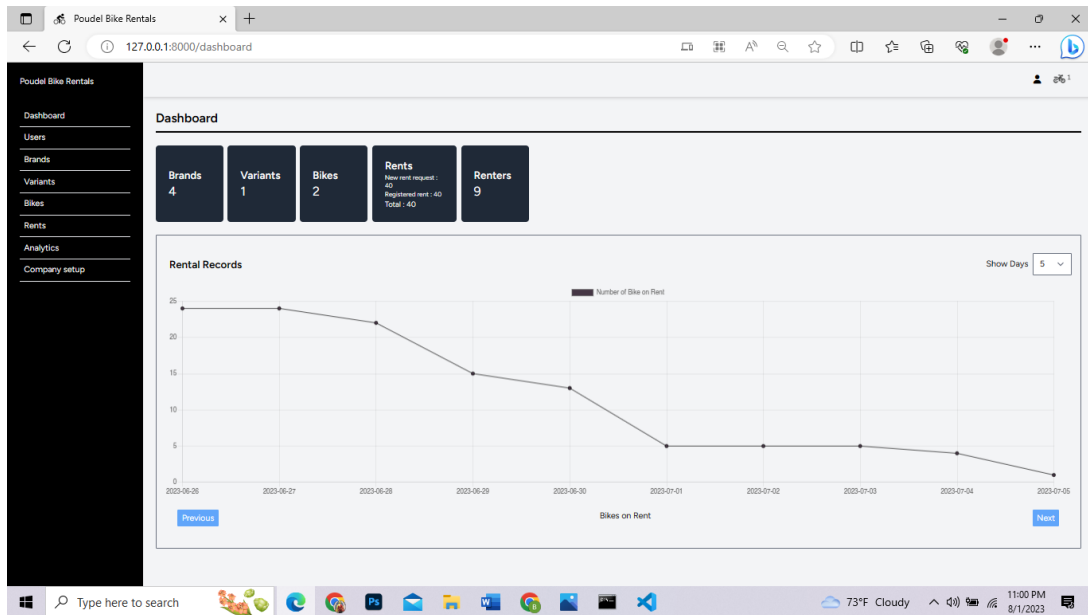


Figure 11: Admin Dashboard

**Poudel Bike Rentals**

Dashboard  
Users  
Brands  
Variants  
Bikes  
**Rents**  
Analytics  
Company setup

### Rents

Show 10 Entries

Select By Payment Method: Show all

Select By Rental Status: Show all

[Add Rent](#)

Sn	Renter Name	Rental Bike	Rental Number	Booking Date	From Date	To Date	Total Rental Price	Payment Method	Rental Status	Action
1	Bipin Paudel	3233	648bc46a932a61	1 month ago	2023-06-23	2023-06-28	Rs 6500	Online	Pending	<a href="#">Change status</a>
2	Bipin Paudel	3233	648bc46a932a61	1 month ago	2023-06-23	2023-06-28	Rs 650	Credit	Pending	<a href="#">Change status</a>
3	Bipin Paudel	3233	648bc46a932a61	1 month ago	2023-06-22	2023-06-28	Rs 7800	Online	Pending	<a href="#">Change status</a>
4	Bipin Paudel	3233	648bc46a932a61	1 month ago	2023-06-22	2023-06-28	Rs 7800	Online	Pending	<a href="#">Change status</a>
5	Bipin Paudel	3233	648bc46a932a61	1 month ago	2023-06-22	2023-06-28	Rs 7800	Online	Pending	<a href="#">Change status</a>
6	Bipin Paudel	3233	648bc46a932a61	1 month ago	2023-06-22	2023-06-28	Rs 7800	Online	Pending	<a href="#">Change status</a>
7	Bipin Paudel	3233	6493d3ea5e4ce1	1 month ago	2023-06-22	2023-06-27	Rs 650	Cash on Hand	Approved	<a href="#">Change status</a>

Figure 12: Rents Table

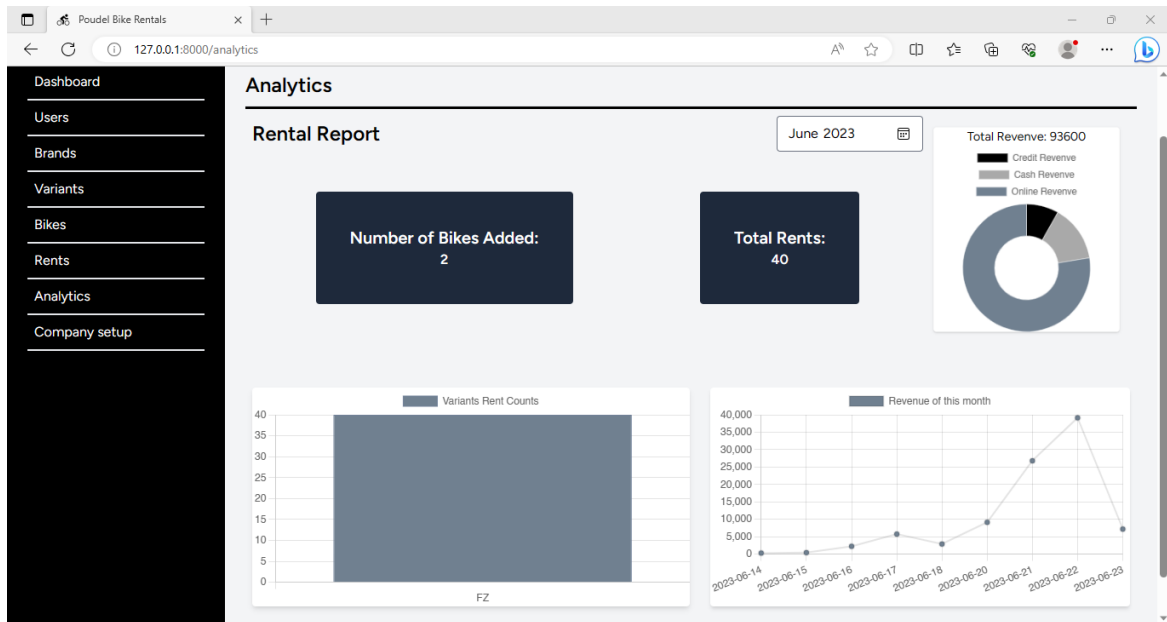
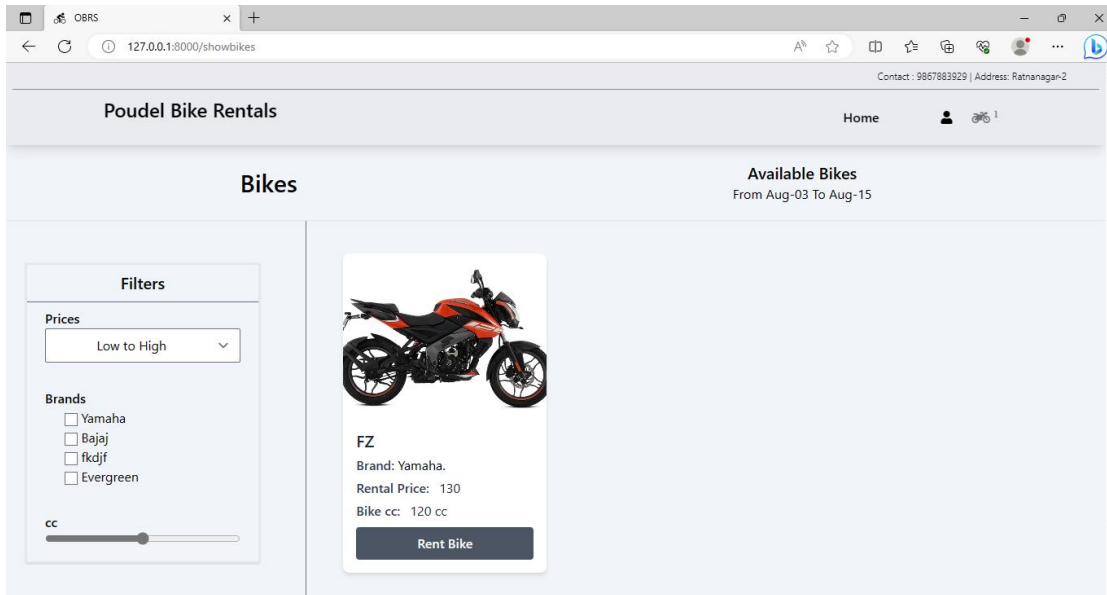


Figure 13: Analytics



**Figure 14: Show Bikes**

## References

- [1] X. a. Z. S. a. Y. S. Gao, An analysis of the current status and countermeasures of bike-sharing in the background of Internet, Chicago: ICVRIS, 2018.
- [2] E. a. U. V. E. Eren, "A review on bike-sharing: The factors affecting bike-sharing demand," *Sustainable cities and society*, vol. 54, no. Elsevier, p. 101882, 2020.
- [3] M. Kramer, "Best practices in systems development lifecycle: An analyses based on the waterfall model," *Review of Business \& Finance Studies*, vol. 9, no. kramer2018best, pp. 77-84, 2018.
- [4] A. a. B. M. K. a. A. A. a. B. A. N. Chakraborty, "The role of requirement engineering in software development life cycle," *Journal of emerging trends in computing and information sciences*, vol. 3, no. chakraborty2012role, p. 5, 2012.
- [5] S. a. M. G. a. S. S. Thorsten and Liao, System Design with SystemCTM, Newyork: Springer Science \& Business Media, 2007.
- [6] H. M. a. Z. A. M. Abdullah, Frontend and backend web technologies in social networking sites: Facebook as an example, IEEE, 2014.
- [7] T. Jindal, "Importance of Testing in SDLC," *International Journal of Engineering and Applied Computer Science (IJEACS)*, vol. 1, no. jindal2016importance, pp. 54-56, 2016.