Project 13

Jot Bikes - Electric Bike Rentals & Sales Rental Management System

About the Client:

Jot Bikes is an Electric Bike & Scooter retailer based in Redfern, Sydney.

Jot Bikes is Australia's Number 1 electric bike, electric scooter, sale, repairs and rental shop. We offer the highest quality range of E-Bikes and E-Scooters with Fast Shipping Australia-wide and excellent customer service.

We specialise in a range of brands including Dragon, Kristall, NCM, The Cullen, Vamos, Bolzzen, Dulatron, Inokim, Kaabo, Mercane, Segway, Vsett, Xiaomi, Zero and many more!

Our products are shipped fast from our shop in Redfern, Sydney. You can visit our shop to browse the range, get repairs, rent an e-bike or e-scooter and get helpful advice.

Project Brief:

❖ **Modules:**

The system comprises of 20 major modules with their sub-modules as follows:

1 User Registration & Login Module
2 User Profile Management Module
3 Bike Catalogue Module
4 Bike Rental Management Module
5 Bike Sales Management Module
6 Search & Filter Module
7 Booking & Payment Module
8 Inventory & Stock Management Module
9 Admin Dashboard Module
10 Feedback & Rating Module

11 Report Generation Module
12 Notification & Reminder Module
13 Maintenance & Service Tracking Module
14 Employee Management Module
15 Discounts & Offers Module
16 Data Backup & Recovery Module
17 AI-Based Bike Recommendation Module
18 AI Chatbot Support Module
19 Predictive Maintenance & Analytics Module
20 Demand Forecasting & Insights Module

❖ Advantages

- Get bikes and scooters easily on rent

❖ Limitation

- Data need to be entered properly otherwise outcome may wont be accurate.

❖ Application

- This system can be used by the multiple peoples to get the counselling sessions online.

Functional Requirements

Functional requirements are product features or functions that developers must implement to enable users to accomplish their tasks. So, it's important to make them clear for the stakeholders. Generally, functional requirements describe system behavior under specific conditions.

The developers of this system must enhance the performance and efficiency of the system by adding 15 to 20 more functional requirements. Students need to do their own research to find how they can improve the system and which FRs need to added. The group must need a prior approval from the stakeholders/project supervisor before finalizing these Functional Requirements.

These enhanced FRs must be reflected separately in Final SRS Report after the approval.

User Module

1. The system should allow new users to register using email and password.
2. The system should verify user email before account activation.
3. The system should allow users to log in and log out securely.
4. The system should allow users to update their personal information and password.
5. The system should display user rental and purchase history.

Bike Catalogue

6. The system should display a list of all electric bikes with details such as brand, model, price, battery range and image.
7. The system should allow users to view detailed information for each bike.
8. The system should support sorting bikes by price, popularity or rating.
9. The system should allow users to filter bikes by category (rental/sale), battery capacity or power output.

Bike Rental

10. The system should allow users to select a bike for rental.
11. The system should allow users to choose rental start and end dates.
12. The system should calculate rental cost based on duration and model.
13. The system should check bike availability before confirming booking.
14. The system should send confirmation emails for successful bookings.
15. The system should allow users to cancel rental bookings before start date.
16. The system should track ongoing rentals and mark bikes as unavailable.
17. The system should record return date and automatically calculate total charges.

Bike Sales

18. The system should allow users to add bikes to a shopping cart.
19. The system should allow users to complete a purchase via integrated payment gateway.
20. The system should generate invoices for completed sales.
21. The system should allow admin to update stock levels after sales.
22. The system should store customer purchase records in the database.

Admin Module

23. The system should allow admin to log in via a secure admin portal.
24. The system should allow admin to add, edit or delete bike listings.
25. The system should allow admin to update rental pricing structures.
26. The system should allow admin to view and manage all user accounts.
27. The system should allow admin to view all completed rentals and sales transactions.
28. The system should allow admin to generate daily, weekly and monthly reports.
29. The system should allow admin to manage employee accounts and roles.
30. The system should allow admin to update service and maintenance records.
31. The system should allow admin to view system analytics (sales trends, rental patterns).

Payment & Notification

32. The system should support multiple payment options (credit/debit, PayPal).
33. The system should securely process and validate payments.
34. The system should email payment receipts to customers.
35. The system should send automatic reminders for upcoming rental returns.

Feedback & Support

36. The system should allow users to post feedback or report issues.
37. The system should allow users to rate bikes after purchase or rental.
38. The system should display average ratings for each bike.
39. The system should allow admin to view and moderate user reviews.

Maintenance & Reports

40. The system should log each maintenance activity for a bike.
41. The system should track next service due date for each bike.
42. The system should generate reports summarising revenue and usage.
43. The system should allow export of reports in PDF or Excel format.

Security & Backup

44. The system should automatically back up data weekly.
45. The system should restrict admin-only modules using role based access control.

Smart Recommendation and Insights

46. The system should use a simple AI algorithm (e.g. content-based filtering) to recommend electric bikes to users based on their browsing and booking history.
47. The system should suggest the Top 3 Recommended Bikes on the homepage using popularity and user ratings.
48. The system should display Frequently Rented Together suggestions based on past rental data.
49. The system should analyse user preferences (e.g. average rental duration, preferred model) and provide personalised offers.

Chatbot and Customer Support

50. The system should include an AI-powered chatbot to answer common user queries such as rental duration, pricing, or bike specifications.
51. The chatbot should guide users through the booking or purchasing process interactively.

52. The chatbot should automatically escalate complex or unanswered questions to the admin dashboard for manual response.

Predictive Maintenance and Analytics

53. The system should use historical service data to predict when a bike might require maintenance.
54. The system should alert admins when a bike's mileage or rental frequency indicates probable wear or battery degradation.
55. The system should generate monthly Bike Health analytics reports using rental, feedback and service data.

Fraud Detection and Anomaly Alerts

56. The system should monitor abnormal booking patterns (e.g. repeated last-minute cancellations) and alert the admin for review.
57. The system should validate input data (rental duration, payment amount) and flag anomalies using basic rule based logic.

Demand Forecasting

58. The system should analyse past rental and sales data to forecast high-demand models for the next month.
59. The system should provide admin with a dashboard graph showing predicted rental demand trends.

Integration Notes for AI Features

- All AI features can be implemented using simple PHP logic and MySQL queries, supported by:
    - ✓ Pattern analysis scripts for recommendations.
    - ✓ Rule-based chatbot implemented via PHP or JavaScript.
    - ✓ Predictive alerts based on threshold logic or basic regression formulas.
- No external machine-learning libraries are required

Non Functional Requirements

There are a lot of software requirements specifications included in the non-functional requirements of the system which contains various processes, namely Security, Performance, Maintainability and Reliability.

Performance

1. The system should respond to user requests within 3 seconds under normal load.
2. The system should support at least 100 concurrent users.
3. Database queries should execute within 2 seconds for common operations.

Usability

4. The system interface should be intuitive and mobile responsive.
5. All major functions should be accessible within 3 clicks.
6. The UI should include visual indicators for availability and booking status.

Reliability

7. System uptime should be at least 99% excluding scheduled maintenance.
8. Data should not be lost in the event of a sudden power failure.

Security

9. User passwords must be encrypted using SHA-256 or bcrypt.
10. SQL injection and XSS vulnerabilities must be prevented.
11. Session timeout should occur after 15 minutes of inactivity.
12. Payment data should be transmitted over HTTPS with SSL certificates.

Maintainability

13. The system should use modular PHP code following MVC structure.
14. Database should use relational integrity and normalisation (3NF).

15. Source code should be documented for developer maintainability.

Portability

16. The system should run on Windows, Linux, or macOS servers with PHP ≥ 8.0 and MySQL ≥ 8.0.
17. The web interface should be compatible with all major browsers.

Scalability

18. The system should allow adding new modules such as bike insurance or subscription plans without major redesign.

Backup & Recovery

19. Database backups should be stored securely and retrievable within 24 hours.

Compliance

20. The system must comply with Australian data privacy laws (APP 2022).

Hardware Requirement: Should be recommended by the developers.

Software Requirement: Should be recommended by the developers.