Biker’s Portal

1. **Overview:**

**Problem Statement:**One of the key shared problems of Bike Enthusiasts is the lack of centralized information. Enthusiasts, buyers, renters, researcher are often constrained by a lack of easy access to reliable detailed specifications, reviews, comparisons and maintenance tips about different bike models. The scattered information on many websites and forums contributes to the hassle.

**Solution :**  
The Biker's Portal is a desktop application based on java designed to act as a one-stop center for all information on bikers. Which serves as a "Wikipedia for Bikes", the application will provide in-depth specifications, features, historical insights, fuel consumption, maintenance tips such as servicing schedules, troubleshooting guides and expert advice, comparisons of two bikes based on performance, cost, mileage,etc. , user reviews and show room and road side price comparison. The interface will be easy to use and enable the users to browse, search and eliminate the issue of dispered and unreliable data.

**Working :**

* **Application Launch:** On the launching of the application, the main dashboard will load, offering a menu that has options such as bike categories, searching for particular bikes, and their details.
* **User Interaction Flow:**
  + **Browse Bikes:** Bikes can be browsed based on type, brand, and price range. The application will fetch information from the database and display it in an easily understandable format, which contains images, specifications, and reviews.
  + **Search Functionalities:** Users can input keywords such as bike model, engine capacity, and price in the search bar to find specific bikes. The application will query the database based on the user input and return relevant results.
  + **Comparison:** A dynamic tool which will allow users to compare two bikes at a time.
  + **Maintenance Tips:** General maintenance guidelines and care tips to keep the bike in top condition will be provided in detail.
* **Database Management:**
  + All data (bike specifications, reviews, maintenance tips) will be stored in a relational database like MySQL.
  + The application will use Java Database Connectivity (JDBC) to connect with the database.
* **Interactive User Interface:**
  + The UI will be built using Java Swing to create a visually appealing and user-friendly interface.
  + Key elements are the menu bar, search bar, result display area.
* **Data Updates & Synchronization:** The admin panel would update the database or the system syncs the information with the cloud database, provided that there are plans to provide online access.

**Advantage :**

* Centralized Information : All information at one application.
* User-Friendly : Easy Navigation for smooth browsing and searching.
* Time-Saving : Eliminates the need for search websites.
* Reliable Data : Ensures data is verified and consistently updated.
* Comparisons : Enhances decision making with AI comparisons.

**Disadvantage :**

* Initial data collection and updating may be time-intensive.
* High dependency on users contributions for their reviews.
* The desktop – only application may lack mobility compared to mobile apps or web platforms.
* Storage – intensive if a large amount of bike data is included.

**Objective :**

The Biker's Portal will be the premier centralized hub for all bike-related information, providing a comprehensive, reliable, and user-friendly solution for bike enthusiasts, buyers, renters, and researchers. This Java-based desktop application addresses the current challenges faced by bikers, which include the fragmented and scattered nature of information across various websites and forums. Consolidation of such detailed specifications, reviews, comparisons, and maintenance tips into a single platform not only increases the accessibility but makes bike data more reliable.

The objectives are:

* Centralized hub : Develops an application, well structured platform offering detailed bike specifications, fuel consumption, maintenance tips, price range, reviews.
* Bike Comparisons : Implementing a dynamic comparison tool of two bikes based on parameters like performance, mileage, cost, fuel type, fuel consumption.
* Maintenance & Troubleshooting : Offers step-by-step maintenance tips, service schedules and troubleshooting solutions to different bike models.
* Decision-making for buyers : Provide real time user reviews, opinions, community discussions to help buyers to choose the right bike.
* Data accuracy & reliability : Regularly updating information using verified source, manufacturer data and user insights.
* User Engagement : Develop interactive features like Q&A forums, expert blogs, AI based recommendations for personal user experiences.
* Data Management : We will create front-end by java swing and backend by JDBC(Java Database Connectivity) and MYSql to fetch and update data.

**Scope :**

* **Global Reach**: Can be expanded to cover international bike models.
* **Integration with Dealers & Service Centers**: Providing direct connections for servicing and purchasing.
* **Mobile App Compatibility**: Future scope includes a mobile application for easy access.
* **Real-Time User Engagement**: Live forums, Q&A sessions, and expert blogs.
* **AI-Based Recommendations**: Personalized bike recommendations based on user preferences.

1. **Project Scheduling (Time, Resource, Allocation, Work Done)** :

**Front-end : Java Swing**

**Jyoti Gupta : Main Dashboard & browsing/search UI**

* + - Main Dashboard :
      * Menu Bar with options like browse bikes, search, comparison.
      * Home Screen UI displaying bike categories with icons / images.
    - Browsing & Search UI :
      * Browsing Features:
        + Browse bikes by brand, type and price range.
        + Display list of bikes with images and brief specifications.
      * Search Functionality :
        + Search Bar allowing users to find bikes based on model name, fuel type, engine capacity and price range.
        + Filters results dynamically and display in JTable/list format.

**Kashish : Comparison**

Comparison UI:

* + - * UI for side-by-side two comparison screen.
      * Drop-down or selection method for choosing two bikes.
      * Display comparison results in table format, including parameters like mileage, engine power, fuel consumption & price.

**Backend : Java + MySQL + JDBC**

**Dayanand : Database Design & Core Data Handling**

* + - Database Design (MySQL):
      * Create table for bikes(model, brand, price, engine, fuel, type, etc), specifications(mileage, power, fule, comsumption, etc), comparisons (bike a v/s bike b)
      * Define primary key, foreign keys and relationships.
      * Ensure proper data normalization for efficient queries.
    - Data Fetching & Storage (JDBC) :
      * Establish JDBC connection between Java Swing & MySQL.
      * Implement methods to retrieve and store bike specifications.
      * Handle CRUD operation for bike details. (Create, Remove, Update, Delete)
    - Search & Filtering Logic :
      * Process user search queries dynamically.
      * Apply filters for price range, fuel type, engine capacity and brand.
      * Optimize queries for fast search results.

**Aryan : Comparison**

* + - Comparison Functionality :
      * Retrieve two bikes specifications from the database.
      * Implement logic for side-by-side comparison based on : mileage, engine power, fuel type, cost, performance.
      * Format and return results in a structured table.
    - Comparison Display & Processing :
      * Develop logic for calculating differences between bikes.
      * Display results in sorted order based on key performance metrics.
      * Ensure real-time updates when a user selects different bikes for comparison.

**Summary :**

Jyoti Gupta – Main Dashboard UI

Browsing UI

Search UI with filters

JTable Results

Kashish Maurya – Comparison UI

Dropdown selection

Dynamic Specification Display

Dayanand Gawade – Database Schema Design

JDBC Connection

CRUD Operation

Search & Filtering Logic

Aryan Tambe – Comparison Logic

Processing

Real-time Structured Display

1. **Requirement Analysis (Functional & Non-functional) :**

**FUNCTIONAL REQUIREMENTS :**

* + User Interface :
    - Main dashboard with menu options (browse bikes, search, comparison).
    - Home Screen with bike categories displayed with images.
    - Browse bikes by brand, type and price range.
    - #Search bar allowing keyword searches for bikes (model name, engine capacity, price, etc).
    - Comparison tool for side-by-side bike comparisons.
    - Display comparison result in tabular format.
  + Data Management :
    - Store bike specifications, maintenance tips and comparisons
    - in a MySQL database.
    - Establish JDBC comparison between Java Swing Application
    - and MySQL database.
    - Implement CRUD operations (Create, Read, Update, Delete)
    - For bike data.
  + Browsing and Search :
    - Fetch & Display bike information from the database based
    - On user inputs.
    - Implement search functionality with filters (price range, fuel
    - Type, engine capacity, brand)
    - Dynamically filter and display search results in JTable or list
    - format.
  + Comparison Functionality :
    - Retrieve specifications for two bikes from the database for
    - comparisons.
    - Display side-by-side comparison results based on
    - parameters (mileage, engine power, fuel consumption,
    - price).
  + Maintenance Tips :
    - Provide detailed general maintenance guidelines and care tips.

**NON-FUNCTIONAL REQUIREMENTS :**

* + Usability :
    - User-friendly and visually appealing interface using java swing.
  + Performance :
    - Optimize database queries for fast search and comparison
    - results.
    - Ensure quick loading times for different sections of the
    - application.
  + Reliability :
    - Ensure data accuracy by regularly updating information
    - From verified sources.
    - Implement errors handling to manage database connectivity
    - Issues.
  + Scalability :
    - Design the application to handle a large amount of bike data.
    - Plan for potential future expansion to include international bike models.
  + Security :
    - Ensure secure storage and transmission of data.
  + Maintainability :
    - Write clean and modular code to facilitate easy maintenance and updates.
    - Document the code and provide clear instructions for future
    - developers.
  + Future Enhancements :
    - Plan for the integration of mobile app compatibility for easy access.
    - Consider implementing AI-based personalized bike recommendations.
    - Enable real-time user engagement features such as live forums and Q&A sessions.
    - Add admin panel for updation.

1. **System Design :**

* ERD :
* Class :
* Use Case :
* Sequence :
* State Transition :
* Deployment :

1. **Development :**

GitHub :

Interface(UI/UX) :

1. **Testing(only test cases no theory) :**

Unit

Integration

System

Acceptance

1. **Future Scope :**
2. **Reference :**
3. **Plagiarism :**