Camera Accessories Renting Platforms (CARP)

Project Planning:

* + We are building a web platform where photographers can:
* List their camera accessories for rent.(With photo of their accessories and detail description of the condition of the accessories.)
* Search for accessories and rent accessories.
* Manage their rentals and bookings.
* Complete payments for rentals.
* Rate and review each other based on rental experiences.
  + Important Pages:
* Home Page
* Login/Logout Page
* Register Page
* Accessory Listing Page
* Search Results Page
* User Dashboard
* Checkout and Payment Page
  + Database Schema: Note: Model mhanje (a special kind of object – it is saved in the database)
* **User Model**: To store user information.
* **Accessory Model**: To store accessory details (e.g., name, description, price, availability).
* **Rental Model**: To track the booking details (e.g., dates, user, accessory, price).
* **Transaction Model**: To handle payment details and track successful payments.
  + Core Features and Functionalities:
    - Authentication System:
* Set up user registration and login using Django.
* Implement password reset functionality.
* Add email verification for user signup.
* Create profile management: users can update personal information like profile picture, bio, and contact details.
  + - Accessory Listing:
* Create a model for accessories (name, description, price, availability, location, owner).
* Develop a form for users to add and edit accessory listings.
* Enable image upload for accessories
* Status of the accessory (is\_available, booked)
* Create accessory listing.( display accessories in list format, and allow users to click on an accessory to view details.)
  + - Search and Filters:
* Implement a search bar using Django.
* Create filter options for:
* Location: Users can input their location or select from a dropdown.
* Price Range: Users can filter accessories based on their rental price per day.
* Accessory Type: Use categories like lenses, tripods, cameras, etc.
* Display the search results dynamically
  + - Booking System:
* Create a booking model that stores information about the accessory, renter(jo udhar ghet aage tyo), rental start and end dates, and booking status (e.g., booked, completed, cancelled).
* Develop a booking form that lets users select the rental period and request a booking.
* Implement booking availability checks: ensure that the accessory is available during the selected period.
* Notify the lender(jo udhar det aahe tyo) when a booking request is made (via email).
* Add a booking management section where users can view their active and past rentals(udhari).
* Implement cancellation policies (allow users to cancel bookings before a certain time).
  + - Payment System:
* Integrate a payment gateway like Razorpay to process payments.
* Create a payment model to store transaction details (amount, payment status, accessory, user).
* Develop a checkout flow: once a user confirms a rental, they proceed to payment.
  + - Reviews and Ratings System:
* Create a review model jeki “rental and accessory models” sobat link asen.(rating, comments, user).
* Display reviews on the accessory’s detail page.
  + - User Profiles:
* Develop a profile page where users can view their details, past rentals, and reviews received.
* Add a section where user can manage accessories listed for rent.
* Include a history of transactions and rental bookings.
* Enable profile picture upload and bio editing.
  + - Admin Dashboard(already build-in in Django):
* Use Django’s built-in admin panel and customize it to manage users, accessories, rentals, and transactions.
  + Tools
* Frontend tools : HTML/CSS/JS/ OR Use Angular
* Backend Tools: Django, third-party Libraries/Packages
* Database Management: SQLite
* Payment Gateway: Razorpay
* Cloud Storage for Media (Accessory Images): (We will need a place to store the images of accessories that users upload.)
* Version Control: Git and GitHub