**Project Name**

**VastraCare** (*Vastra* = clothes in Sanskrit).

**1. Core Features**

**User Side**

* **User Registration & Login** (Email/OTP, Google Sign-In)
* **Dashboard**
  + Track laundry status
  + View reminders
  + Subscription details
* **Smart Reminders**
  + Predictive alerts for laundry
  + Event-based reminders
* **AI Stain Detection**
  + Upload photo of stained garment
  + Get treatment suggestions
* **Place Laundry Order**
  + Choose services: wash, iron, dry clean, stain removal, shoe cleaning
  + Select pickup & delivery times
* **Subscription Plans**
  + Weekly, biweekly, monthly
  + Add-on services & fragrance options
* **Payment Integration**
  + Razorpay / Stripe / PayPal
* **Order Tracking**
  + Real-time updates on pickup, washing, and delivery
* **Ratings & Reviews** for laundry service providers

**Service Provider Side (Local Washer / Laundry Professional)**

* **Profile Creation** & Verification
* **Order Management** (accept, update, complete)
* **Earnings Dashboard**
* **Schedule Management** (availability slots)

**Admin Side**

* Manage users & service providers
* Approve service provider registrations
* View analytics (orders, revenue, employment created)
* Manage subscription plans & pricing

**2. Tech Stack**

**Frontend (Web/App)**

* **React.js** (Web)
* **React Native** or Flutter (Mobile App)

**Backend**

* **Node.js + Express.js** (scalable and fast)

**Database**

* **MongoDB** (flexible for orders and user data)

**AI Integration**

* TensorFlow.js or a custom API for stain detection (can train a model or use a pre-trained vision API like Google Vision)

**Cloud Storage**

* Cloudinary or AWS S3 (for images)

**Authentication**

* Firebase Auth or JWT (JSON Web Tokens)

**Payments**

* Razorpay (India) or Stripe (global)

**Notifications**

* Firebase Cloud Messaging (push notifications)

**Deployment**

* Backend: Render / Railway / AWS EC2
* Frontend: Vercel / Netlify / Play Store & App Store
* Database: MongoDB Atlas

**3. Database Structure (Simplified)**

**Users**

pgsql

CopyEdit

user\_id

name

email

phone

address

role (user / provider / admin)

subscription\_plan

wallet\_balance

**Orders**

java

CopyEdit

order\_id

user\_id

provider\_id

services\_selected

pickup\_time

delivery\_time

status (pending / in-progress / completed)

price

payment\_status

**ServiceProviders**

css

CopyEdit

provider\_id

name

phone

address

services\_offered

availability

ratings

earnings

**Reminders**

bash

CopyEdit

reminder\_id

user\_id

type (predictive / event-based)

date\_time

message

**4. Development Phases**

**Phase 1: Planning & Design**

* Wireframes for User, Provider, and Admin dashboards
* Decide brand identity (logo, colors, typography)

**Phase 2: Backend API Development**

* User authentication & role-based access
* CRUD for orders, providers, subscriptions
* Payment integration
* Reminder scheduler

**Phase 3: Frontend Development**

* User dashboard
* Service booking flow
* Provider dashboard
* Admin dashboard

**Phase 4: AI Integration**

* Build or integrate stain detection model
* Connect model to order form for recommendations

**Phase 5: Testing & Deployment**

* Unit tests & bug fixing
* Deploy backend, frontend, and mobile app

**Phase 6: Launch & Feedback**

* Limited beta testing in one city
* Collect feedback from users & providers
* Improve UI/UX and features

**5. Extra Unique Add-Ons**

* Gamification badges for eco-friendly laundry habits
* Donation/Clothes swap section
* Scent customization
* Clothing lifecycle tracking