**Project Overview**

**1. Project Purpose**

The website will serve as:

* **Digital storefront** for Sridhar Chary & Shiva services.
* **Community resource hub** with **live job notifications** (Govt + Private + Remote).
* **Monetization platform** via Google Ads.
* **Role-based system** (Admin + Manager) with analytics.

**2. Key Roles & Responsibilities**

**Manager Role (Highest Level)**

* Full control of the website.
* **Analytics Dashboard**:
  + Ad impressions, clicks, CTR.
  + User visits, page views, session times, device info.
* Manage Admin accounts (add/remove).
* Approve or reject jobs submitted by Admins.

**Admin Role**

* Post new jobs manually.
* Approve/reject jobs fetched from external sources (RSS, APIs, Scraping).
* Edit or delete job posts.
* Manage job categories and content.

**Users / Visitors**

* View services offered.
* Browse live job alerts with search/filter options.
* Interact with Google Ads for monetization.

**3. Frontend (UI/UX)**

**Tech Stack:**

* HTML5, CSS3, JS
* Bootstrap 5 + Tailwind CSS
* Vercel deployment (serverless frontend)

**Pages/Sections:**

1. **Home Page**
   * Hero banner + tagline
   * Service highlights
   * Live job alert ticker
   * Google Ads banner
2. **Services Page**
   * Grid/accordion layout listing all services:
     + LIC Insurance Advisor, Money Transactions, ATM
     + Job Applications & Results
     + PAN / Aadhaar / PF services
     + Ticket Booking (Bus / Flight)
     + Utility Bill Payments (DTH, Gas, Electricity, Credit Cards)
     + Printing / Scanning / Lamination / Telugu Letter Typing
     + Project Reports (Excel / PowerPoint / Word Editing)
     + Vehicle & Sand Bookings
3. **Job Notifications Page**
   * Live job listings from **RSS feeds + Scrapers + APIs**
   * Filters: Category, Location, Job Type (Govt / Private / Remote)
   * Search functionality
4. **Admin Dashboard**
   * Add/Edit/Delete jobs
   * Approve/reject API-fetched jobs
   * View job statistics
5. **Manager Dashboard**
   * All Admin features
   * Analytics: Ads + User engagement (page visits, clicks)
   * Manage Admin accounts
   * Backup/Restore database
6. **Contact Page**
   * Contact form (submits to Flask backend)
   * Prop info: Sridhar Chary & Shiva, Mobile: 9160037232

**4. Backend (Flask + PostgreSQL)**

**Tech Stack:**

* Flask (Python) deployed on **Render**
* PostgreSQL database for structured storage

**Database Schema (Core Tables):**

1. users – (id, name, email, password\_hash, role [admin/manager], created\_at)
2. jobs – (id, title, category, description, link, source [manual/API/scrape], status [pending/approved], posted\_by, created\_at, expires\_at)
3. ads – (id, ad\_type, impressions, clicks, date)
4. analytics – (id, user\_ip, page\_visited, timestamp, device, referrer)

**Backend Features:**

* **Authentication & Role-based Access Control** (Flask-Login / Flask-Security)
* **Job Management:** Admin-approved jobs + API / RSS / Scraping integration
* **Analytics Tracking:** User visits + ad interactions
* **API Integration:**
  + **RSS Feeds:** PIB, Employment News, SarkariResult
  + **Scraping:** TSPSC, APPSC, District Collectorates (Telangana & AP)
  + **REST APIs:** Adzuna, Jooble, Remotive (Private & Remote jobs)

**5. Job Data Sources**

| **Method** | **Sources** | **Notes** |
| --- | --- | --- |
| RSS Feeds | PIB, Employment News, SarkariResult | Structured XML, easy parsing with feedparser |
| Web Scraping | TSPSC, TSLPRB, APPSC, APSSDC, District Collectorates | For state-level jobs; use BeautifulSoup |
| Free REST APIs | Adzuna, Jooble, CareerJet, Remotive | Private + Remote jobs; JSON API |

**Workflow:**

1. Fetch jobs → store in DB (status=pending)
2. Admin approves → job goes live on frontend
3. Manager monitors analytics & ad performance

**6. Monetization & Analytics**

* **Google AdSense:** Display ads in Home & Jobs pages
* **Analytics:**
  + Track user behavior and ad interactions
  + Manager can view CTR, ad impressions, and top pages

**7. Deployment Architecture**

Frontend: Vercel (Static HTML/CSS/JS, Bootstrap + Tailwind)

↓

Backend: Flask (Render Deployment)

↓

Database: PostgreSQL (Render Managed DB)

↓

Job Sources:

• RSS Feeds (Govt Jobs)

• Web Scraping (State + District Jobs)

• REST APIs (Private / Remote Jobs)

↓

Role-Based Dashboards:

• Admin (job management)

• Manager (analytics + admin control)

↓

Google Ads & Analytics

**8. Deliverables**

1. Fully responsive **frontend** (Vercel)
2. Flask **backend** with PostgreSQL (Render)
3. **Dynamic Job Notifications** section with multi-source integration
4. **Role-based system**: Admin & Manager
5. **Analytics Dashboard** for Manager
6. **Google Ads Integration**