



K.R. MANGALAM UNIVERSITY

THE COMPLETE WORLD OF EDUCATION

Recognised under the section 2 (f) of the UGC Act 1956



Empowering the Youth; Empowering the Nation





K.R. MANGALAM UNIVERSITY
THE COMPLETE WORLD OF EDUCATION

MattersUrSkills

Pre-Final Year Project Synopsis Submitted by

ROLL No.	NAME
2401201167	Khushi Chauhan
2401201127	Shweta Jha
2401201162	Akanksha Kumari
2401201217	Vikram
2401201160	Kartik Malhotra

Faculty Mentor: Mohd. Shadav

Project Overview

MattersUrSkill is a MERN stack–based web application designed to connect work providers with skilled and unskilled individuals seeking flexible employment opportunities. The platform focuses on empowering students, housewives, and unemployed individuals by providing access to home-based, part-time, and local service jobs.

The system allows workers to register their skills and availability, while work providers can post task requirements such as job type, location, and payment details. Secure authentication, user verification, and a rating system ensure trust and transparency between users.

By eliminating middlemen and promoting direct communication, MattersUrSkill enables fair opportunities, skill recognition, and economic independence using modern web technologies.

Specific Objectives

- ❖ To design and develop a web-based platform that connects work providers with skilled and unskilled workers.
- ❖ To enable students, housewives, and unemployed individuals to access flexible work opportunities based on their skills and availability.
- ❖ To implement secure user authentication and role-based access using JWT.
- ❖ To provide location-based job matching for local and nearby service opportunities.
- ❖ To eliminate middlemen by allowing direct communication between workers and work providers.
- ❖ To ensure trust and transparency through user verification, ratings, and feedback system.
- ❖ To promote skill recognition and economic independence using modern web technologies.



Key Features

❖ Dual User Registration

Separate registration for workers and work providers.

❖ Skill-Based Work Listing

Workers can list practical skills such as home-based work, local services, and part-time tasks.

❖ Job Posting System

Work providers can post job requirements with location, duration, and payment details.

❖ Location-Based Matching

Connects users with nearby work opportunities for faster and convenient hiring.

❖ Secure Authentication

User login secured using JWT-based authentication.

❖ Direct Communication

Enables direct interaction between workers and work providers without middlemen.

❖ Rating & Feedback System

Ensures trust, transparency, and service quality.

❖ User Verification

Verified profiles to reduce fake users and improve platform reliability.



Project Usecases & Scope

❖ Use Cases :

- ❖ **Students** – Can earn part-time through home-based or nearby jobs.
- ❖ **Housewives** – Access skill-based work from home, e.g., cooking, handicrafts, tailoring.
- ❖ **Unemployed Individuals** – Get verified local jobs or freelance tasks.
- ❖ **Work Providers** – Can find reliable workers quickly without middlemen.
- ❖ **Small Businesses** – Hire temporary help for delivery, repair, or other services.

❖ Scope :

- ❖ Flexible, inclusive, and location-based employment opportunities.
- ❖ Focuses on trust, transparency, and eliminating middlemen.
- ❖ Potential to expand to a mobile app and integrate payment gateways.
- ❖ Could scale for urban & semi-urban areas, eventually nationwide.



Prototype/Project Flow diagram/Architecture

- ❖ **Frontend:** React.js – Interactive UI, responsive design.
- ❖ **Backend:** Node.js + Express.js – Handles API requests, business logic.
- ❖ **Database:** MongoDB – Stores users, jobs, ratings, and logs.
- ❖ **Security:** JWT + Bcrypt – Authentication & password security.
- ❖ **Flow:**

User registers → Login & role selection → Post job / Add skills → Job matching → Direct communication → Task completion → Ratings & feedback.



Data & Resources

Data Used :

- ❖ User profiles: Name, contact, skills, role.
- ❖ Job posts: Description, location, payment, duration.
- ❖ Ratings & feedback: Task quality & user trust.

Resources :

- ❖ **Software:** VS Code, Node.js, MongoDB Compass, Postman
- ❖ **Libraries:** React.js, Express.js, Mongoose, JWT, Bcrypt
- ❖ **Hardware:** Any system with internet and modern browser
- ❖ **Team Resources:** 5 members with divided roles: Frontend, Backend, DB, UI/UX, Deployment



Methodology

- ❖ **User Registration & Authentication** – Secure login with JWT.
- ❖ **Role-Based Access** – Worker or Provider.
- ❖ **Job Posting & Skill Listing** – Data stored in MongoDB.
- ❖ **Location-Based Matching** – Connects nearby users.
- ❖ **Communication** – Secure messaging between users.
- ❖ **Task Execution & Feedback** – Ratings maintain trust.
- ❖ **GUI Interaction** – React frontend for smooth user experience.



Expected Results & Impact

❖ Expected Results :

- ❖ Seamless connection between workers and providers.
- ❖ Verified and trusted job executions.
- ❖ Flexible earning opportunities for students, housewives, and unskilled users.
- ❖ Transparent rating and feedback system.

❖ Impact :

- ❖ Promotes economic independence and skill recognition.
- ❖ Reduces unemployment and reliance on middlemen.
- ❖ Encourages women empowerment through home-based work.
- ❖ Provides a scalable solution for local employment needs.



Project Timeline

❖ Week 1–2: Planning & Design

All Members: Requirement gathering, finalize features and modules.

UI/UX Designer: Create wireframes & mockups for main pages (Worker Dashboard, Provider Dashboard, Job Listing).

Team Lead: Set up GitHub repo, branch strategy, decide tech stack.

Deliverables: Project flowchart, architecture diagram, UI prototypes.

❖ Week 3–4: Backend + Database

Backend Developer: Set up Express.js server, API routes (login, register, job post, job apply).

Database Specialist: Create MongoDB schemas (Users, Jobs, Ratings).

Team Lead: Integrate backend and database, test APIs with Postman.

Deliverables: Functional APIs for user registration, job posting, matching.

❖ Week 3–5: Frontend Development

Frontend Developer: Build React components: login/register, dashboards, job list, profile pages.

UI/UX Designer: Review UI, ensure responsive design, animations.

Team Lead: Connect frontend with backend APIs.

Deliverables: Interactive user interface with real-time data fetching.

❖ Week 5–6: Integration & Testing

All Members: Integration of frontend and backend.

Team Lead: Unit testing, bug fixing, validate API responses.

Database Specialist: Ensure correct data flow and schema integrity.

Deliverables: Fully functional web app with end-to-end flow.

❖ Week 6–7: Features & Security

Backend Developer: JWT authentication, secure passwords with Bcrypt.

Frontend Developer: Add form validations, error handling, notifications.

UI/UX Designer: Final polish of UI, ensure usability.

Team Lead: Implement rating and feedback system.

Deliverables: Secure platform with verified users, task completion, and ratings.

❖ Week 7–8: Deployment & Finalization

Team Lead: Deploy frontend (Netlify/Vercel) and backend (Render/Heroku).

All Members: Final testing, cross-browser checks, bug fixes.

UI/UX Designer: Prepare PPT, flowcharts, screenshots for presentation.

Team Lead: Final review, GitHub repo cleanup, documentation, and synopsis prep.

Deliverables: Live application, ready presentation, final documentation.



THANK YOU

