A Synopsis Report of mini project

On

WorkEase – An On-Demand Domestic Help Platform

For

Partial Fulfillment of Award of the

B. Tech Degree in Information Technology

Under the Supervision of

Dr. Krishan Kumar

Submitted by:

Anay Agrawal (2301920130033)
Aniket Gupta (2301920130035)
Animesh Singh (2301920130036)
Ayush Yadav (2301920130053)

Session: 2025-26

GL Bajaj Institute of Technology and Management,
Greater Noida

TABLE OF CONTENTS

INTRODUCTION

In the fast-moving urban lifestyle, people often struggle to find reliable and skilled domestic help for their everyday tasks such as cleaning, cooking, plumbing, electrical repair, babysitting, and other personal services. At the same time, millions of semi-skilled and unskilled workers face unemployment or under-employment due to the lack of proper visibility, certification, and digital platforms to showcase their skills.

WorkEase is designed as an intelligent, user-friendly, and secure platform that bridges this gap. It connects households and service seekers with verified service providers (maids, electricians, plumbers, beauticians, etc.) through an easy-to-use web and mobile application.

The system enables users to book domestic services in real time, view service provider profiles, ratings, and availability, and make digital payments seamlessly. Simultaneously, it provides employment opportunities to workers by giving them digital visibility, fair wages, and flexibleschedules.

The concept draws inspiration from the operational model of UrbanCompany but aims to focus more on employment generation and transparency for blue-collar workers. The platform incorporates technologies such as React for a dynamic frontend, Flask/Node.js for the backend API, Firebase/MySQL for secure data storage, and Google Cloud for deployment and scalability.

TECHNOLOGY USED

Component	Technology	Purpose
Frontend	React JS, HTML, CSS, JavaScript	Responsive, interactive UI
Backend	Flask / Node.js	Handles logic, authentication, APIs
Database	Firebase / MySQL	Stores data securely
Cloud Platform	Google Cloud / AWS	Scalable hosting and uptime
Authentication	Firebase Auth / JWT	Secure login and encryption
Payment Gateway	Razorpay / Stripe API	Handles payments
Data Analytics	Python (Pandas, Scikit-learn)	Analyzes usage trends
Version Control	Git & GitHub	Team collaboration

PROPOSED WORK

The proposed work for WorkEase focuses on developing an integrated digital platform that connects domestic service seekers with verified household workers in a simple, transparent, and efficient manner. The system aims to organize the informal domestic help sector by providing a structured online environment where users can easily book services and workers can receive stable employment opportunities.

The platform will be designed with three core components: User Application, Service Provider Application, and Admin Dashboard. These components will collectively support service browsing, booking, worker verification, payment handling, and management operations through a modern and scalable web architecture.

Objectives of the Proposed Work

Provide a centralized platform offering various domestic services such as cleaning, cooking, plumbing, electrical repair, babysitting, and more.

Enable service providers to register, showcase skills, upload certifications, and receive job requests directly.

Implement a rating and review system to ensure reliability, transparency, and quality control.

Integrate secure digital payment gateways and automatic invoice generation.

Utilize real-time location tracking to match nearby available workers with customers efficiently.

System-Level Features

Real-Time Matching Algorithm

A rule-based matching system will allocate workers to user requests based on distance, ratings, availability, and skill compatibility.

Digital Payments & Wallet System

The platform will integrate payment gateways (Razorpay/Stripe) for fast and secure cashless transactions and maintain transparent earnings for workers.

Secure and Scalable Architecture

Using React for frontend, Flask/Node.js for backend, Firebase/MySQL for database management, and Google Cloud for deployment ensures reliability, scalability, and efficient data handling.

Expected Outcomes

A functional web platform enabling smooth booking of household services

Enhanced job opportunities for domestic workers

Increased trust and transparency through ratings, verification, and secure payments

Reduced dependency on informal middlemen

A scalable solution that can expand into new cities and additional service categories

METHODOLOGY / EXPERIMENTAL WORK

The development process for **WorkEase** follows the **Agile Model**, allowing iterative and incremental enhancements. The system architecture and workflow are outlined below.

System Architecture

The WorkEase architecture consists of three major layers:

1. **Frontend Layer:**
    - Developed using React, providing users and workers with separate dashboards.
    - Handles UI rendering, API calls, and routing between pages.
2. **Backend Layer:**
    - Built with Flask or Node.js (Express).
    - Includes APIs for user authentication, booking management, payment handling, and notifications.
3. **Database & Cloud Layer:**
    - Uses Firebase for real-time updates and MySQL for relational storage.
    - Data hosted on Google Cloud ensures reliability, backup, and horizontal scaling.

Functional Modules

1. User Module:

Allows customers to sign up, log in, browse available services, select a service provider, book a time slot, and make payments.

2. Service Provider Module:

Workers can register, list their skills, upload verification documents, accept or reject booking requests, and view payment summaries.

3. Admin Module:

The admin monitors platform activity, verifies worker credentials, resolves disputes, and ensures policy compliance.

4. Analytics Module:

Uses Python libraries to analyze service demand trends, customer satisfaction, and worker performance.

System Workflow

The entire workflow proceeds as follows:

1. **Registration:**
   Both users and service providers register via the application using email or mobile authentication.
2. **Profile Verification:**
   Service providers submit identity proofs and skill certificates. Admin verifies and approves them.

3. **Service Browsing & Booking:**
   Users browse categories (cleaning, cooking, electrical, plumbing, etc.) and choose based on location, rating, or price.

4. **Real-Time Allocation:**
   The system matches nearby available service providers using geolocation and time-slot algorithms.

5. **Payment Processing:**
   Users make online payments via integrated gateways.

6. **Service Completion & Feedback:**
   Once the service is completed, users rate the provider. Feedback updates the provider's overall score.

7. **Data Analysis:**
   Analytical scripts track peak demand, average service time, and satisfaction metrics to improve future matching.

Security and Privacy Considerations

- All user data is encrypted using **HTTPS** and **JWT tokens**.
- Role-based access ensures separation of user, worker, and admin privileges.
- Payments are processed through **secure, PCI-DSS-compliant gateways**.
- Personal details of workers are hidden until a confirmed booking to ensure privacy.

CONCLUSION

After implementing the core modules, a functional prototype of **WorkEase** demonstrates the following capabilities:

1. **Interactive User Interface:**
   Users can sign up, search for services, and book providers intuitively.

2. **Dynamic Worker Dashboard:**
   Workers receive job notifications, accept/reject options, and earnings insights.

3. **Admin Control Panel:**
   Centralized management for verifications, analytics, and complaint resolution.

4. **Analytics & Visualization:**
   Data visualizations highlight most-booked services, top-rated workers, and location-wise demand.

5. **Automation & Notifications:**
   The system automatically sends confirmations, reminders, and invoice details via email/SMS.

The experimental results reveal that using digital algorithms for task allocation significantly reduces idle time for workers and increases user satisfaction by minimizing wait time.

Moreover, the analysis module can predict which services (e.g., cleaning or appliance repair) will experience demand surges during weekends or festivals, helping the platform manage resources efficiently.

In conclusion, WorkEase not only digitalizes household services but also acts as a **social impact project**, improving livelihoods, boosting trust in informal labor markets, and contributing toward sustainable urban development.

FUTURE SCOPE

The **WorkEase** project successfully establishes a digital ecosystem that empowers both domestic service providers and service seekers. By creating a transparent and efficient medium, it helps workers gain continuous employment while ensuring users receive trustworthy, affordable, and on-time service.

The system demonstrates how modern web technologies and data-driven insights can solve socio-economic challenges like unemployment and unorganized labor.

Key Achievements

- Developed a functioning prototype with secure login, service listings, and booking management.
- Enabled real-time worker allocation and digital payment integration.
- Provided analytical insights into demand and customer behavior.

Future Enhancements

1. **AI-Based Recommendation Engine:**
   Use machine learning to recommend suitable workers based on previous bookings, ratings, and skill match.
2. **Mobile App Deployment:**
   Extend the system to Android/iOS using React Native for better accessibility.

3. **Multilingual Support:**
   Integrate local language options for ease of use among semi-literate workers.

4. **Subscription & Loyalty Programs:**
   Offer monthly service plans for households and reward points for frequent users.

5. **Skill Development Integration:**
   Collaborate with NGOs or government programs to provide digital literacy and skill enhancement to workers.

6. **Automated Dispute Resolution:**
   Include AI chatbots and NLP-based complaint handling to improve service response.

REFERENCES

- https://react.dev/
- https://nodejs.org/
- https://flask.palletsprojects.com/
- https://firebase.google.com/
- https://cloud.google.com/
- https://razorpay.com/docs/
- https://developer.mozilla.org/en-US/docs/Web
- https://scikit-learn.org/stable/
- https://urbancompany.com/