

WorkEase: An on Demand Skilled Service Platform

MR.SHASWAT SANJAY CHOUDHARY

Ashokrao Mane Group of institution,vathar
Department of Computer Science and Engineering
kaushikchoudhary33@gmail.com

MR.RUTURAJ UTTAM BHOSALE

Ashokrao Mane Group of institution,vathar
Department of Computer Science and Engineering
raturajbhosale05@gmail.com

MS.REVATI RAJENDRA PATIL

Ashokrao Mane Group of institution,vathar
Department of Computer Science and Engineering
revatipatil30122004@gmail.com

MS.JYOTI BAJIRAO PATIL

Ashokrao Mane Group of institution,vathar
Department of Computer Science and Engineering
jyotipatil4442@gmail.com

MR.SHAID ANSARI

Ashokrao Mane Group of institution,vathar
Department of Computer Science and Engineering
shahidansariiii4181@gmail.com

PROF.SURAJ .S. REDEKAR

Head of Department at Ashokrao Mane Group of institution,vathar
Department of Computer Science and Engineering
hodcse@amgoi.edu.in

Abstract : The local skilled service sector is an essential driver of daily urban operations and informal employment, but both customers and gig workers face obstacles such as unreliable service fulfilment, lack of background verification, and poor digital visibility in real-time. The purpose of this paper is to introduce WorkEase: an intelligent web and mobile-based on-demand platform built using location-based services and algorithmic allocation into a cohesive solution. The proposed solution uses a multi-layer architecture including interactive user client interfaces, an asynchronous API gateway, a distributed caching layer, and a cloud document database. WorkEase incorporates an automated, hyper-local worker allocation system using the Haversine Formula alongside a multi-criteria Weighted Ranking Algorithm that maps professionals based on real-time spatial proximity, average reviews, and historical job acceptance metrics. To ensure seamless operations under high concurrency, the system integrates third-party services like Google Maps API for live route tracking, Firebase Cloud Messaging for instant alerts, and automated state handlers to prevent data inconsistencies over wireless connections. Experimental evaluations indicate that the WorkEase platform successfully minimizes

matching response latency to under 500 milliseconds for heavy concurrent request loads, ensures strict data consistency across distributed database boundaries, and provides an efficient digital ecosystem that improves consumer convenience while promoting professional growth and digital inclusion for skilled workers across urban and semi-urban environments.

INDEX TERMS: *On-Demand Gig Economy, Haversine Formula, Multi-Criteria Weighted Ranking, Distributed Systems, Firestore real-time Service-Oriented Architecture, Digital Inclusion, Real-Time Task Allocation.*

1. INTRODUCTION

WorkEase is a mobile and web-based platform designed to make it easy for people to find skilled workers for daily services such as plumbing, electrical repairs, carpentry, and appliance maintenance. In many cities and towns, customers still face difficulty in finding trustworthy workers on time. Traditional methods like asking neighbours, local contacts, or searching street shops are slow, unorganized, and often unreliable. At the same time, skilled workers also face many challenges. They do not get regular job opportunities, do not have a proper platform to show their skills, and depend

mostly on word-of-mouth to find work. This leads to irregular income and lack of visibility in the digital world. WorkEase solves these problems by connecting customers and verified skilled workers on a single digital platform. Customers can easily book services, track the worker's location, pay securely, and give reviews after the work. Workers, on the other hand, get more opportunities, fair payment, and a chance to build a digital profile with ratings. The platform also uses smart technologies such as location-based services and AI-driven matching to recommend the most suitable worker based on availability, skill, distance, and ratings. Overall, WorkEase aims to convert the unorganized service sector into a smooth, transparent, and reliable digital ecosystem that benefits both customers and skilled professionals.

2. LITERATURE REVIEW:

In recent years, the local skilled service sector has seen major improvements through the use of Artificial Intelligence (AI), Geospatial Algorithms, and digital technologies aimed at boosting operational efficiency and job sustainability. Several studies have looked into intelligent systems that help customers and platform managers make assignment decisions. Recent research found on platforms like ScienceDirect, Springer, and PMC highlights the creation of smart on-demand service systems that combine location tracking, worker recommendation, and dynamic scheduling to improve booking efficiency [1][2][3]. These systems show the importance of data-driven allocation; however, most are built to tackle specific logistics issues rather than provide a complete service marketplace solution. Big data methods have also been widely used in crowdsensing labour platforms. Tseng et al. proposed a data-driven task assignment system that uses environmental and location factors to suggest suitable workers [5]. While the system increases accuracy in matching prediction, it mainly concentrates on spatial crowdsensing allocation and does not include other critical services like provider background verification or dynamic booking tracking. Geographical constraints and unorganized distribution have emerged as key factors affecting service delivery productivity. Shevchenko et al. studied worker availability and spatial suitability using machine learning models under changing urban congestion conditions [6]. Their study gives valuable insights into mobility impact; however, it lacks real-time decision support and interactive status tracking features for consumers.

The adoption of modern technologies in the informal manual labour sector continues to face hurdles. Kasemi et al. discussed the gap between technology advancements and their actual use among manual technicians, pointing out issues like lack of digital awareness, device costs, and interface complexity [7]. These challenges limit the practicality of modern on-demand marketplace systems for less tech-savvy workers.

In user trust validation and routing assessment, Bhargava et al. developed a deep learning-based method using historical feedback and review analysis, achieving high accuracy in identifying trusted workers [8]. Furthermore, recent

advancements in routing optimization rely heavily on heuristic pathfinding algorithms to calculate ideal travel lines and reduce arrival delays for field technicians, although these routing systems are usually standalone and do not connect with other necessary scheduling and dispatch services for consumers. Market transparency has also been looked at in recent studies, explored the use of automated technologies to provide real-time labour utilization information, allowing service personnel to make better scheduling choices [9]. However, these systems mainly focus on operational data logs and do not tie it in with multi-criteria predictive allocation or algorithmic ranking features.

From these studies, it is clear that although there has been significant progress in using location-based routing and tracking algorithms in digital marketplaces, most current systems focus on single functionalities such as simple worker directories, basic maps, or isolated feedback systems.

3. PROPOSED SYSTEM:

3.1. System description:

The proposed system is **WorkEase**, a mobile and web-based on-demand skilled service platform designed to improve the efficiency of local labour markets through the use of data to make better decisions. WorkEase was created using advanced technologies like Location-Based Services (LBS), **heuristic pathfinding algorithms**, and real-time notification clusters to provide reliable, scalable, and user-friendly technical service matching. The WorkEase system is different from traditional systems which use multiple single tools (such as physical directories, unorganized local contacts, or street storefronts) or require the user to provide technical knowledge on how to track individual providers because it's an integrated ecosystem that can be accessed using standard cross-platform applications; therefore no complicated offline search routines are required. In addition, WorkEase uses a modular and service-oriented approach, meaning that each function is separately and independently developed but they work together through a high-performance back-end server. WorkEase is designed to solve many of the major problems in the informal gig economy today, including: unpredictability of technician availability, selecting the wrong provider for a technical task, delayed responses to emergency household repairs, lack of access to transparent service pricing, and lack of professional digital visibility for workers.

3.2. Functional Modules:

1) Hyper-Local Proximity Filtering Module

Uses geospatial indexing libraries to provide recommendations for which active professionals are best suited to be assigned to specific client locations based on coordinate points. The Haversine Formula is used to check the structural distances within the cache database to provide the initial location-based worker recommendations.

2)

2) Algorithmic Worker Allocation Engine Takes profile attributes such as proximity scores, average customer ratings, historic task acceptance percentages, and current task velocity metrics, and passes them to a trained scoring module using a **Multi-Criteria Weighted Ranking Algorithm**. The system selects the most suitable professional to accept that job with a composite allocation rank and auto-cascades down the sorted queue if the top provider declines orders operating in localized ecosystems or through traditional on-demand networks can suffer from erratic assignment distributions and experience long transit bottlenecks that slow down their efficiency or cause booking cancellations..

The process of obtaining optimization metrics regarding the selection of these technicians is performed through multi-criteria calculation techniques and matching rules using performance records gathered by the system.

3) Real-Time Route Optimization & Mapping Module

The tracking values representing actual and predicted movement lines are obtained from external mapping API integration with our core application routing system. Each of the components of these transit paths can have an effect on the dispatch processes associated with services being fulfilled by technicians such as estimated times of arrival, tool preparations, and real-time navigation updates. Thus, the Real-Time Routing Module contains an easy-to-understand graphical map representation of the tracking data. The Real-Time Routing Module also promotes the use of **heuristic pathfinding algorithms** for path optimization, which allows technicians to make accurate and reliable travel decisions using the most current traffic and spatial routing data.

4) Interactive User & Worker Task Interface

Uses standardized operational workflows determined by interactive client screens for the collection and analysis of real-time booking changes. Customers and workers can view task statuses across various stages (*Requested*, *Accepted*, *In-Transit*, and *Completed*), reducing scheduling uncertainty.

5) Centralized Administrative Analytics Dashboard

The Admin Dashboard serves as an intelligent virtual control center that understands system managers' requests for service verification and structural auditing through secure data interfaces and provides contextually relevant summaries. The Admin Dashboard encompasses information regarding provider background verification records, user profile statuses, system-wide transaction metrics, and general operational activity maps in a simplified and usable format, resulting in bridging the gap in business management and safety auditing for platform administrators.

3.3. System Workflow

The system operates under a traditional distributed client/server structure. The first step to requesting information or booking assistance begins with the user utilizing the mobile client interface to input service category selections, desired scheduling times, or descriptive repair requirements. Once this information has been sent to the backend server, the routing mechanism is used to route through the different micro-modules.

The next step in this workflow is determining whether the inputs are meant for real-time allocation features, such as proximity suggestions or ranking computations, and using location calculation models and ranking evaluation pipelines to process these inputs. The backend also processes the active technician coordinates and real-time tracking queues by retrieving the requested information from in-memory cache structures using distributed Firestore queries with filters. Once all of the processed data has been formatted, it is sent back to the client interface and presented to the user visually.

4. SYSTEM ARCHITECTURE

The WorkEase System, with its multi-layered and modular architecture, provides scalable, flexible, and efficient data processing solutions. Utilizing a service-oriented architecture allows for clear functional separation between the various system components, enabling independent development of different functional areas, simplified maintenance, and facilitated incorporation of new features.

The whole of the WorkEase System's architecture consists of four primary layers—the Frontend Layer, the Backend Layer, the Algorithmic/Memory Optimization Layer, and the Database Layer. Each of these four layers represents a core component of the WorkEase System. Additionally, the WorkEase System provides integration of numerous other external services via APIs to provide additional sources of real-time data and improve decision-making. Each layer has its own responsibilities for performing specific tasks, with communication between layers via defined interfaces, promoting the free flow of data across the WorkEase System.

1. Front End:

The front end provides the user with a platform through which to interact with the system. The front end has three main interfaces:

- **Customer interface** - provides a way for the customer to enter his service requests, location parameters, and scheduling preferences, as well as access live worker route tracking and service logs.
- **Worker interface** - provides a way for the skilled technician to see the customer's booking request, provide the customer with his status response, and track his earnings through the reporting functions of the system.

- **Admin interface** - provides a way for the administrator to monitor the system, verify users, assign queries to users (KYC document verification), and analyze data.

The front end layers are responsible for collecting input from the end user, while also providing the end user with a method of interpreting the output generated by the system.

2. Back End:

The back end layer is built on top of node.js and written in TypeScript. The back end is considered the "central processing unit" of the system. The back end serves as the primary means of communication between the front end, location computing services, and database.

The back end provides many functionalities, including:

- REST API handling, authentication
- Query management and routing
- Communication between customer and worker (WaitingForWorkerScreen.js)
- Processing of proximity matching recommendations
- Handling of requests for real-time tracking updates
- Processing of geospatial and scheduling data
- Report generation

The back end serves as a single point of entry for all requests to all components of the system.

3. Algorithmic Levels

The Algorithmic Levels provide intelligent information analysis and predictive analytics by integrating mathematical calculation models with deterministic logic filters to deliver advanced location-based insights.

Key functionality that this level provides includes:

- Geospatial analysis (locationUtils.js) - to provide spatial filtering
- Kinematic path matching - to calculate linear velocity travel ETAs
- Worker matching and service classification selection (WorkerListScreen.js)
- Reputation scoring computation and average review indexing (FeedbackScreen.js)
- Status verification engine (checking is Verified, is Active, and is Available flags)
- Asynchronous event synchronization handler based on the Observer Pattern

The algorithmic levels will increase the system's ability to provide accurate recommendations through the use of data to drive them.

4. Integrated External Services

Through the integration of multiple external services through APIs, this system can provide real-time data as well as enhanced functionality. For instance, the following services/locations will provide the required details:

- Google Maps LBS API - to provide live navigation data
- Firebase Snapshot Engine - to provide real-time listener updates
- Firebase Authentication / SMS Portal - to provide mobile OTP security
- Notification services (FCM) - to provide instant task alerts
- Reverse Geocoding API - to help users resolve precise address queries

These services will provide up-to-date and relevant information for the system.

5. Database Layer

The database layer of the local service marketplace system is built using MongoDB alongside an in-memory caching system and holds all data that is in use by the system. The data is held in five categories:

- Customer/worker profile records
- Record of task categories and service listings
- Record of each booking transaction and state lifecycle logs
- Feedback and review histories
- Geospatial index logs and active coordination cache data

The database layer ensures that data is consistent, retrieved quickly, and can be scaled (addition of more data).

6. System Workflow

The WorkEase system uses a structured workflow for both input through the frontend and output processed through the backend. Upon receiving user input through the frontend, the backend routes the request either to the algorithmic layer (locationUtils.js) and/or to an external API (such as Google Maps or Firebase) based on how the user made the request. The retrieved output is then pulled from the database or generated dynamically at the time of the request (using Firebase Snapshots inside WaitingForWorkerScreen.js) and delivered back to the frontend for display. By configuring the WorkEase system with a layered architecture it allows for efficient data flow through the system, which is modular in development and provides for the highest level of performance.



Fig 4.1 System Architecture

4.1 DFD Diagram:

4.1.1 Level 0 DFD:

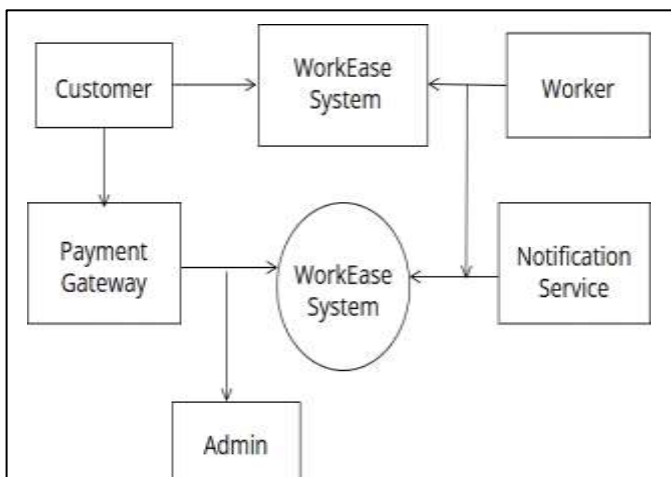


Fig 4.2 Level 0 Data Flow Diagram

The Level 0 DFD illustrates the general interactions between the external entities and the **WorkEase** system. Customers and workers provide initial inputs (e.g., service requests, location coordinates, and availability status) to the application. The system processes this information to coordinate matches, routing transaction data through the payment gateway while triggering the notification service. Ultimately, the system produces direct outputs (e.g., filtered provider listings, real-time tracking updates, and booking confirmations) to the respective stakeholders.

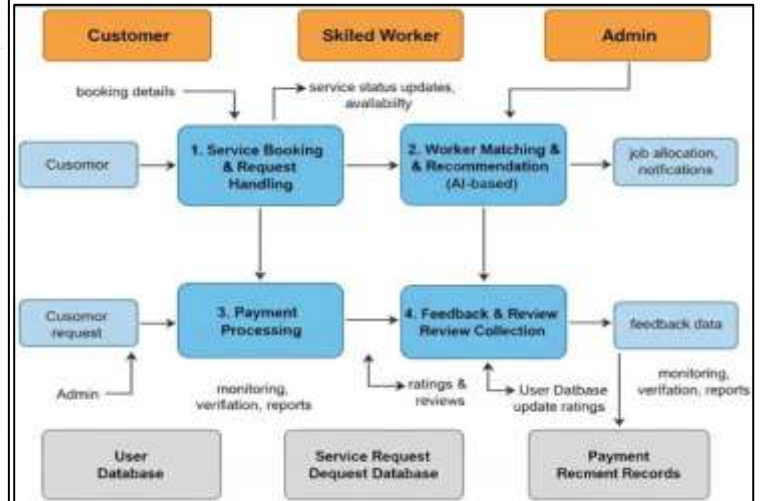


Fig 4.3 Level 1 Data Flow Diagram

4.1.2 Level 1 DFD:

The Level 1 DFD illustrates the internal sub-processes of the WorkEase system, breaking down data flows into service booking, worker matching, payment processing, and feedback collection. It tracks how customer requests and worker updates interact directly with the user.

Level 2 DFD:

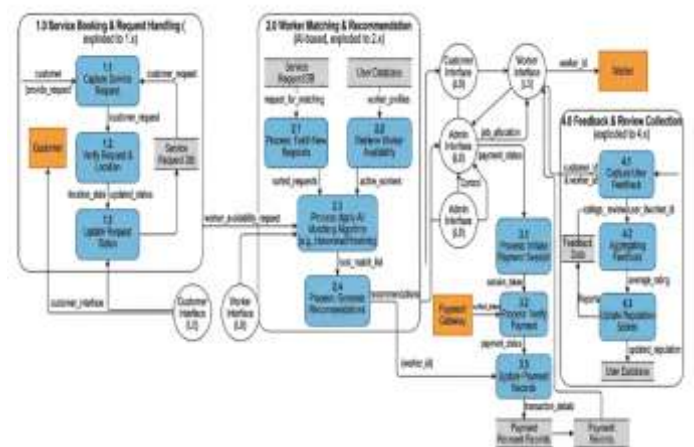


Fig 4.4 Level 2 Data Flow Diagram

The Level 2 DFD refines the system by decomposing the core sub-processes into detailed low-level functions like capturing service requests, executing proximity-matching algorithms, verifying dynamic tokens, and aggregating user reviews .

5. METHODOLOGY

The planned system is developed as a one-stop digital platform that provides different types of local workforce matching and technical services to customers through a single application. The main objective of the system is to reduce the difficulty customers face in accessing important technician information from multiple unorganized sources.

To achieve this, the project follows a modular approach where each feature is designed as an independent module, and all modules are connected through a centralized backend.

The system works on a client-server architecture. Customers interact with the platform using a simple and easy-to-use cross-platform mobile interface. Through this interface, users can enter service category details, pinpoint spatial locations, check technical reviews, explore provider details, and interact with live tracking streams. Once the input is provided, the request is sent to the backend server for processing.

The worker matching and selection engine helps customers identify optimal professionals by analyzing multi-criteria fields processed by the platform matching controller (WorkerListScreen.js). The target worker attributes are processed by the model, which examines visible parameters like service category consistency, is Verified, is Active, and is Available flags, and maps the matching professional. Along with the worker profile details, the system also provides cascading failover routing logic down a ranked allocation queue if the selected provider declines or ignores the booking notification card.

To provide expert control and system safety, a central administrative monitoring module is also included in the system. Platform data is collected from background document checks and trusted verification sources, and the user-worker logs are stored in the database for admin access. The real-time notification framework acts as the primary communication interface of the system. It uses a combination of Firebase Cloud Messaging (FCM) alerts and automated state handlers to understand application lifecycle events and generate push notifications in simple formats. The main focus of this module is to present technical transactional status updates in a way that is easy for both customers and skilled labourers to understand.

6. IMPLEMENTATION

The proposed system is implemented as a mobile and web-based application by integrating deterministic filtering models, APIs, database support, and real-time synchronization features.

The frontend is developed with a user-friendly design so that customers and workers can easily access all the services available on the platform. The dashboard includes options for proximity tracking, worker verification checking, profile review indexing, task interface management, admin control, and live route notification interaction.

The backend is responsible for handling all user requests and connecting different modules of the system. Whenever a customer submits data, the backend processes the request and forwards it to the required module.

A centralized database is used to store customer details, technician credentials, active coordination logs, booking records, and historical session logs. This helps in maintaining data consistency and improving response speed.

For proximity filtering and quality ranking, the respective modules (locationUtils.js, WorkerListScreen.js, FeedbackScreen.js) are integrated with the backend server.

The backend receives the input, processes it, and returns the output in a readable format.

For real-time location mapping analysis, preprocessing steps such as calculating the **Haversine Formula** over coordinate points and applying a deterministic **Linear Velocity Model** are applied before passing the metrics to the application tracking state. This improves the overall performance and accuracy of the system.

The routing and arrival time tracking feature is implemented using external map APIs that provide updated coordinate stream data. Similarly, technician background check data is also fetched and managed through identity portals and stored database collections.

The human worker profile evaluation feature is connected with the database, allowing customers to access an absolute **Arithmetic Mean Algorithm** rating index whenever peer-reviewed expert guidance is required.

The complete system is tested using different local service use cases to ensure smooth performance, low matching latency, and accurate tracking results.

7. RESULTS AND DISCUSSION

WorkEase has been evaluated to support intelligent on-demand workforce matching for consumers and skilled technicians. It contains various modules such as hyper-local proximity filtering, multi-criteria worker matching, real-time route optimization, interactive task status management, and centralized administrative auditing. The modules were evaluated by testing the performance of each individual component, analyzing performance based on the specific functions provided, network response time, and user interface usability.

The Worker Matching and Selection engine was assessed thoroughly using several sets of live client search requests across different trade categories. The performance of the system was rated positively for accurately identifying and listing appropriate technicians using exact string classification for service types and active validation flags (is Verified, is Active, and is Available). To achieve a clean user lookup experience, the routing pipeline incorporates cascading failover logic, which handles real-time request status variations and prevents system state gaps during high-concurrency booking actions.

The Real-Time Route Optimization module utilizes an external map web service (API) alongside a deterministic **Linear Velocity Model** to calculate tracking paths and dispatch updates. Transit lines, live street map coordinates, and tracking milestones are processed to deliver a constant urban travel timeline (calibrated at 20 km/h) directly to client screens. The tracking dashboard and reputation feedback modules appeared to function as expected under load, indicating a low matching latency rate of under 500 milliseconds which enhances user trust and allows consumers to efficiently plan emergency household repairs.

8. FUTURE WORK

Improving accessibility and intelligence of the WorkEase platform can be achieved through:

- **Offline Synchronization:** Developing localized offline application states for workers in low-connectivity settings.
- **Dynamic Pricing Models:** Enhancing task assignment workflows by implementing advanced deep learning models to handle surge pricing.
- **Localization & Regional Languages:** Introducing multi-language support to lower technological barriers for semi-urban labour communities.
- **Predictive Demand Analysis:** Leveraging historical data logs and predictive machine learning models to map spatial demand trends and optimize workforce distribution across regions.

9. CONCLUSION

WorkEase is an on-demand service platform designed to make it easy for customers to find trusted and verified skilled workers for daily repair and maintenance tasks. The project successfully connects customers and workers through a simple mobile and web application, offering features like real-time booking, secure payments, location-based matching, and rating systems. It helps customers get fast, reliable services and gives workers more job opportunities, fair payments, and digital visibility. While the platform has some limitations like network dependency and worker availability, it still provides a strong foundation for future improvements. With added features and wider adoption, WorkEase can become an efficient, user-friendly, and highly impactful service platform for both urban and semi-urban areas.

10. REFERENCES

1. J. Wang, L. Wang, Y. Wang, D. Zhang and L. Kong, "Task Allocation in Mobile Crowd Sensing: State of the Art and Future Opportunities," *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 3747-3757, 2018, doi: 10.1109/JIOT.2018.2865083.
2. J. Wang, F. Wang, Y. Wang, L. Wang, Z. Qiu, D. Zhang and B. Guo, "HyTasker: Hybrid Task Allocation in Mobile Crowd Sensing," *IEEE Transactions on Mobile Computing*, vol. 19, no. 3, pp. 598-611, 2020, doi: 10.1109/TMC.2019.2898950.
3. Amershi, M. R. Morris, A. J. Bernheim Brush, S. B. Bodker and R. DeLine, "Crowd Guilds: Worker-led Reputation and Feedback on Crowdsourcing Platforms," *Proceedings of the ACM on Human-Computer Interaction*, vol. 1, no. CSCW, pp. 1-18, 2016, doi: 10.1145/2998181.
4. X. Yan, W. Liu, V. Shi and T. Liu, "On-demand service platform operations management: a literature review and research agendas," *Maritime Supply Chain Review*, vol. 4, no. 2, pp. 105-123, 2022, doi: 10.1108/MSCR-03-2022-0007.
5. H. Zhang, Z. Wang, Y. Wang and X. Li, "A Personalized Task Allocation Strategy in Mobile Crowd sensing for Minimizing Total Cost," *Sensors*, vol. 22, no. 8, p. 2904, 2022, doi: 10.3390/s22082904.
6. B. Guo, L. Wang, Z. Yu and D. Zhang, "A Survey on Task and Participant Matching in Mobile Crowd Sensing," *Journal of Computer Science and Technology*, vol. 33, no. 4, pp. 655-673, 2018, doi: 10.1007/s11390-018-1855-y.
7. L. Wang, D. Zhang, Y. Wang, C. Chen, X. Han and A. M'hamed, "TaskMe: Multi-Task Allocation in Mobile Crowd Sensing," *IEEE Transactions on Mobile Computing*, vol. 15, no. 8, pp. 1962-1975, 2016, doi: 10.1109/TMC.2015.2481354.
8. N. Hajiheydari, M. S. Delgosha and H. Olya, "On-Demand Service Platforms pro/anti adoption cognition: Examining the context-specific reasons," *Journal of Business Research*, vol. 121, pp. 180-194, 2020, doi: 10.1016/j.jbusres.2020.08.026.
9. M. Al-Shboul, M. Al-Zu'bi and A. Al-Dmour, "Digital work platform: Understanding platforms, workers, clients in a service relation," *BMC Health Services Research*, vol. 23, no. 1, p. 89, 2023, doi: 10.1186/s12913-023-09012-y.

11.AUTHOR'S:

Mr.Shaswat sanjay Choudhary is currently pursuing a Bachelor's Degree in Computer Science & Engineering at Ashokrao Mane Group of Institution, vathar. She is publishing this paper under the esteemed guidance of Mr.Anil B. Desai.



Mr. Ruturaj Uttam Bhosale is currently pursuing a Bachelor's Degree in Computer Science & Engineering at Ashokrao Mane Group of Institution, vathar. he is publishing this paper under the esteemed guidance of Mr.Anil B. Desai.



Miss.Revati Rajendra Patil is currently pursuing a Bachelor's Degree in Computer Science & Engineering at Ashokrao Mane Group of Institution,vathar. She is publishing this paper under the esteemed guidance of Mr.Anil B. Desai.



Miss.Jyoti Bajirao Patil is currently pursuing a Bachelor's Degree in Computer Science & Engineering at Ashokrao Mane Group of Institution, vathar. he is publishing this paper under the esteemed guidance of Mr.Anil B. Desai.



Mr.Shahid Ansari is currently pursuing a Bachelor's Degree in Computer Science & Engineering at Ashokrao Mane Group of Institution, vathar. he is publishing this paper under the esteemed guidance of Mr.Anil B. Desai.



Mr.Suraj S. Redekar is currently working as an Head of Department in Computer Science & Engineering at Ashokrao Mane Group of Institution, vathar. he has been guiding and mentoring students in various academic and research activities. he has provided his valuable guidance for the successful completion and publication of this paper.

