International Journal of Research and Development in Applied Science and Engineering (IJRDASE)
ISSN: 2454-6844

WorkWhiz: A User-Centric Approach to Streamlining Local Skilled Worker Connections

Pervez Rauf Khan, Asad Alam, Amir Sohail, Adiba Anwar, Arshiya Akram
Dept of Computer Science Engineering,
Integral University, Lucknow, India
asadalam4291@gmail.com, amirsohail9564@gmail.com, adibaanwar7@gmail.com, arshiyaakram9@gmail.com

Abstract: The project "WorkWhiz" tackles the problem of connecting users to skilled workers (e.g., carpenters, electricians, plumbers) through a mobile responsive web application. Finding a worker through word of mouth or local advertisement is traditionally tedious, and unreliable. This project takes advantage of new web technologies by providing a user-friendly web application with geo-location look-up and category searching for quick retrieval of various workers. Major features consist of catalogs of workers' profiles, responsive design, user-friendly design, and effortless navigation. Although additional components like user registration and authentication, payment gateways, and payment processors will be added later, significant progress has already been made. "WorkWhiz" seeks to decrease the time, attempts, and opportunity barriers to skilled workers, thus, helping service providers and homeowners without setting difficulty, directly through their smartphones. This project shows the potential of thriving on advanced technology to fix gaps set out by the conventional labor market, providing a measurable and smart solution to access available workers.

Keywords: WorkWhiz, Mobile-Responsive Web Application, Skilled Worker Search, Location-Based Search, Worker Categories, User Experience, Responsive Design, React/Tailwind CSS/PHP/JavaScript, Bootstrap Framework, Worker Profiles, Search Filtering, Intuitive Navigation, Future Enhancements, User-Friendly Interface, Home Improvement Services, Construction Industry, Digital Platform, Proximity Search.

1. Introduction

With everything in today's digital age moving so fast, there's never been a higher demand for quality workers in construction, home renovations, and the trades, but finding the right service professional to get the job done has always been a struggle. "WorkWhiz" app comes to the rescue in this aspect by offering a user-friendly web-based platform to find and hire skilled workers like carpenters, electricians, plumbers, etc. at preferred location and according to the specific requirements. Through user-friendly UI and mobile-first approach, workwhiz provides the tools to homeowners, contractors and businesses to quickly find local workers within no time while also reducing the hassle, effort and complexity required for getting a job done.

Smartphones and their accompanying high speed internet services have brought a revolution in user attitudes with an increased focus on convenient solutions and multitasking. Outdated methods for recruiting workers remain insufficient in the modern world and include unautomated personal referrals, posting ads locally or external networks of contractors which do not provide real-time responsiveness. WorkWhiz eliminates such inconvenience by employing slocation-based searches accompanied by category filters which allow users to effortlessly find professionals tailored to their specific tasks. Such approaches based on user experience allow greater accessibility and interaction between employers and employees, fulfilling a significant gap in the labor market.

Despite its benefits, creating such a platform to locate workers is incredibly difficult due to the ever-changing requirements of mobile responsiveness, intuitive user navigation, and organized presentation of worker details. WorkWhiz, however, addresses these issues through careful design process combining the fundamentals of responsive web design, minimalist interface, effortless mobile adjustment of content, and a vision for future updates. Providing an internal framework for advanced elements including user identity verification and the ability to monitor the employee's participation in predefined tasks comes later.

1.1 Challenges in Developing the "WorkWhiz" Application

The "WorkWhiz" Web Application -mobile based web application (application which will run on mobile browsers as well(IT professional) faced various challenges in designing and implementing the mobile enabled web application of "WorkWhiz"- especially with location based search, user experience and scalability to handle future extensions. Here, we describe the key challenges encountered in building the "WorkWhiz" mobile application:

a. Implementing Effective Location-Based Search

The primary feature of 'WorkWhiz' is the ability to search for prospective employees by location. However, supporting an efficient and accurate location-based search has much complexity involved:

• Geolocation Accuracy: It's usually hard to obtain precise geolocation information from your users, especially across browsers and devices. Differences in GPS accuracy or user consent settings can result in inaccurate or unreliable location data.

• Scalability of Search Queries: Serving a large volume of geo-located queries would require faster algorithms and database

indexing. Search results may degrade if not optimized as the number of workers or users increases.
• Location Data Privacy: The seeking users location data protection is very important in building trust relation and privacy law obeying to. Sterile packaging and transport make the system difficult to use.

b. Ensuring Mobile Responsiveness Across Devices
The mobile-first philosophy of `WorkWhiz` demands compatibility with many devices and screen sizes, as well as browsers. This brings its own set of challenges:
• Cross-Device Compatibility: Testing your work across phones, tablets, and desktop computers, all of which must be usable, is quite a chore. Rendering engines and resolution variances could produce unpredictable results or layout.
• Performance Optimization: It will usually be the case that mobiles have lower processing power and available bandwidth than desktops. It means keeping it fast and snappy, especially for very data-heavy worker lists, requires some tweaking of assets such as images, CSS and JavaScript.
• Touch-Based Interaction: If there is a requirement to support touch with near parity to the desktop, where the requirement is also to support later mobile users, the target points are not obvious to create.

c. Effective Management of Employee Information Displaying employee profiles with standard inclusions such as name, skills, and contact information is a basic feature of "WorkWhiz." Displaying and keeping this sort of stuff in a useful format, however, is far from simple in a number of ways:
• Data Consistency: Worker Data like contact details, skills etc should be consistent and recent and that may not be the case if received from static JSON files or a future backend. Manually updates or ineffective data management can even cause inaccuracies to be introduced.
• Scalability for Huge Worker Listings: With the growth of the platform, the database may contain numerous workers. Retrieval and displaying large data without affecting performance needs to be taken care of with proper database design and caching methods.
• Data Security: Protection of employees' personal data, i.e., contact details, from unauthorized use or access is crucial. The absence of a backend in the initial phase makes it more challenging to practice secure data handling.

d. Designing an Intuitive User Interface
The effectiveness of "#WorkWhiz" is highly contingent on developing a user-friendly interface that appeals to all users, regardless of their level of computer literacy. This sets a variety of problems:
• Functionality or Simplicity: It's tough to make a clear and simple interface full of deep filtering and search and vice versa. For some of the best relationships and the best applications, less is more!Too much in the UI can distract, yet too little can suck the life out of functionality.
• Consistent User Experience: There's no way for navigation, search and filtering to be consistent across pages and devices. Poor UX will continue to drive users mad, reducing engagement.
• Accessibility Compliance: Making an interface accessible, which complies to standards like WCAG (Web Content Accessibility Guidelines), involves providing additional features, like support for screen readers, keyboard navigation, and high-contrast modes.

e. Laying Foundations for Future Growth
Although the core functionality built for "WorkWhiz" is limited to simple functionality such as location based searching, directories of worker, this was a sensible decision, since project can be easily expanded to include mechanisms such as authentication for user and rating and so on. So do the longer-range factors:
• Modular Code Design: Hard to have clean, modular and refactoring friendly code in order to be ready for future features (such as backend integration, real time availability) > if you have project deadlines.
• Database Scalability: Using static data (e.g., JSON) at first probably won't work as well for dynamic parts such as user accounts or real-time updates. Using a database backend (Node.js,PHP) at some later date will have to be planned to reduce refactoring.
• Feature Integration Complexity: Adding a complex feature – such as worker rating system, live availability or payment gateway- is bound to add complication, such as users session management, data consistency, and real-time data-flow management.

1.2 Existing Solutions and Their Limitations
Tools and method have been developed and can be extended to help address the user-competent worker matchmaking problem, in particular in the context of the building industry, including construction trades and home improvement trades. Tools and platforms like this are created to streamline the process of finding and hiring workers, but usually have some degree of restriction that impacts their ease of use.

a. Conventional Practices
Old-school methods of finding skilled labourers, like word-of-mouth referrals, searching local ads, and asking around for recommenedations, have long been the lifeblood of hiring in the construction and home renovation industry, includes:
• Accessibility: They are embedded within local networks and can be easily accessed with no need for digital equipment and/or the Internet.
• Trust: Referrals come with a level of trust – they are based on firsthand experience by trusted associates.
• Low Cost: They're typically inexpensive or free, because there are no subscription or platform fees.

Limitations:

• Time-Consuming: Searching for candidates through classified or word of mouth is very time consuming as you make pol and calls.
• Restricted Coverage:The plans are geographically bound, i.e., no employee outside the defined geographical area may be covered.
• Absence of Standardization:There is no single place to look to confirm workers ' credentials, capabilities, and quality, resulting in varying levels of quality.
• Inefficiency: There were no filters or search functions, making it difficult to find employees quickly based on applied skills and availability.

b. Online Classified Platform
Local directories and online classified sites offer online alternatives to the conventional processes, where employers can search through directories to find potential employees in their area.

• Wider Reach: The internet is much more widespread than your local network and you can find workers across a greater area.
• Category-Based Listings: Several platforms enable (and at times require) users to categorize workers (e.g., as plumbers, electricians) in order to facilitate efficient search.
• Ease of Access: The listings are accessible on the web or app through internet connection-- making it accessible to users.
Limitations:
• Lack of Validity: Job offers on employees are usually not accompanied with dumbed down data validity on experience, skills and reliability, making easier to resort to staff of dubious skills.
• Stale Information: The directories can be updated only from time to time, and contact with staff who are no longer available may occur.
• Poor Mobile Optimization: Most classified websites are not properly optimized for mobiles, they offer a poor user experience on mobiles.
• Limited Interactivity: These sites don't usually have a lot of bells and whistles like real-time booking, worker profiles, or chat services.

c. Freelance and Gig Economy Platforms
There are proliferation of gig economy websites such as TaskRabbit and Thumbtack which help people to find skilled workers or freelancers to perform every variety of task from general home improvement to construction.
• Intuitive Interfaces: The websites have intuitive interfaces and have the feature of searching and filtering for skills or locations.
• Worker Profiles:Comprehensive review often includes worker ratings, reviews, as well as work experience to help users make more educated judgments on hiring.
• Mobile Applications: Both hostings have mobile apps for easy access to your job portals when you're on the move.
• Booking and Payment Systems: Some sites integrate booking and payment processing, which makes it easier to hire.
Limitations:
• Cost:These kinds of websites have a service charge or get a commission, so the cost for hiring is higher for the employer and charged to the employees as well.
• Complexity: Complex parts eg bookings, or user logon are overwhelming to users when all they want is quick basic fixes!
• Limited Availability of Labour: Labour may not be available in necessary places or may even be in demand for some of the skilled labour components such as, the rural areas.
• Internet Dependence: Services like these are always dependant on having internet access, which may turn off users in areas with poor internet access.

d. Contractor Networks and Agency-Based Services Contractor networks and agency services provide pre- screened workers or contractors to customers to manage their large projects, such as home renovations.
• Quality Assurance: The workers themselves are screened by the agency so it is more likely that you get someone reliable and professional.
• Specialist Services: These are profession-specific or top-end project-specific networks, where you will find senior professionals.
• Project Management: Additional services such as project coordination or oversight may be offered by agencies as well.
Limitations:
• High Cost: Agency work is expensive, making it less accessible to individual or low-budget work.
• Limited Flexibility: Because agency staff are assigned in order of priority, clients will have fewer choices of who will work for them.
• Geographic Constraints: The services are urban- oriented, there are fewer services for the people living in the villages.
• Time Delays: The agency recruitment can take longer than direct recruitment because of consults and scheduling.

2. Related Work

There has been a significant interest in the development of online platforms that match users with the services of skilled workers in recent years, particularly as a result of the rise of the gig economy and the increasing demand for on-demand services. Many research and development projects have focused on the design and deployment of such platforms, with a focus on user experience, mobile support and efficient search assistance.
Smith et al. [1] designed a location-based service platform for homeowners to search for the nearest contractors. Their paper underlines the importance of geolocation APIs such as Google Maps for the efficient proximity based search in service providers. The platform is built with a mobile-first front-end written in HTML, CSS and JavaScript, which has enabled it to be available cross device. However their system does not allow category based filtering, which limits the ability for users to filter the searches by trades i.e. plumbers or builders. This gap is compatible with the intention of the "WorkWhiz" project to use category-based filtering for greater search precision.

Jones et al. [2] proposed a Git-collaboration-management web application to hire freelance workers, while Using Bootstrap and React frameworks for a better user experience. Their work points to the need for responsive design techniques, such as media queries and flexible grid layouts, to support compatibility across smartphones, tablets, and desktops. Despite -they accept profiles of workers with limited information, their solution does not consider advanced features, such as user verification or the capture of the user availability in real time, as well as the WorkWhiz gaps. Their findings are consistent with the requirement for a mobile- friendly interface, that "WorkWhiz" has fully incorporated to take account of modern user habits.

Similarly, Gupta et al. [3] explored expanding frameworks in service centrals with a focus on cloud-based architecture systems and their qualitative analytical scalability. Their application backend is constructed with Node.js and MongoDB, where the storing and retrieval of worker informat is automated. Their paper demonstrated the gains achieved with location-driven search precision in cloud solutions using APIs for real-time data processing and streaming. However, their application has features like payment portals that are outside the purpose of "WorkWhiz" which seeks to augment the user experience through minimalism devoid of offline contact arrangements and the scoped features. Pertaining to the use of static JSON data in the initial stage of "WorkWhiz", aligns with their suggestion regarding light data management paradigms for early-stage work.

Williams et al. [4] studied user experience guidelines pertaining to service-oriented web applications, paying particular attention to navigation and design simplicity. Their system uses Figma or similar software to create wireframes and mockups of user interfaces to be user-tested using prototyping. Designs are iterative based on user-centered design principles. Their work underscores having a clearly defined mechanism to traverse pages, such as traversing the search box to access worker profiles, which "WorkWhiz" accomplishes through iterative design and testing. However, their site has user account registration, a feature that "WorkWhiz" chose not to include in order to maintain simplicity in the initial phase.

Kim et al. [5] focused on search filtering issues in web applications, particularly in service marketplaces. Their work highlights how search time is minimized by filtering by category and location. They implemented JavaScript-based event handling for dynamic update of search results, which aligns with the strategy of "WorkWhiz" towards interactivity. Their findings suggest that filtering improves user satisfaction, which supports adding search filtering in "WorkWhiz," as it expands the capability to effectively locate desired workers.

Zhang et al. [6] examined the testing strategies of mobile responsive web applications paying particular attention to testing cross-browser functionality and mobile responsiveness. Their study employed tools like BrowserStack to check consistency across devices and browsers. Their findings emphasize the value of iterative testing, which "WorkWhiz" employs. However, their platform has real-time elements, such as worker availability, that "WorkWhiz" purposefully set aside to limit the scope of the project.

Even with these advances, the marketplace for services still faces some challenges. Existing platforms mostly focus on cumbersome features like user registration, payment processing, or real-time monitoring which frustrates users looking for basic options. Additionally, most offer very little effort to enable users to filter by categories which would allow them to easily find specialized workers. The "WorkWhiz" initiative fills these gaps by providing a sleek, mobile-friendly design focused on location and category-based searching while avoiding complex features to enhance usability and scalability. This project builds on the success of previous work by integrating mobile-first design with quick-to-use, intuitive navigation, and effective searching that meets the needs of users looking to hire skilled workers.

3. Theory and Calculation

Here we introduce the theoretical basis and calculations.

3.1 Theoretical Framework

The framework for the "Workwhiz" project is user-centered design, responsive web development, and location-based service systems. This framework seeks to create an effective mobile web app that seamlessly connects users to appropriate workers, contingent on their locations and personalized preferences. The framework's foundational principles include human-computer interaction (HCI), information retrieval, and mobile primary, which are aimed at optimization of availability, usability, and effectiveness.

The basis of the framework has the following key aspects:

a. User-Centered Design Model: The UCD model targets the needs, preferences, and behavior of the end-users during the entire process of development. The model advocates for the iterative design virtualization for the UCD principle which drives collecting feedback from users to enhance the system's interface and functionality. "WorkWhiz" is also guaranteed an easy, intuitive platform through UCD that focuses on user-friendliness for effective task completion, especially for mare-sourcing. The implementation of the model involves wireframing and prototyping (Figma and Adobe XD) for interface design and testing of the design decisions.

b. Location-Based Search Model: This model uses geospatial information to provide pertinent search results based on the user's location in comparison to available workers. This model is based on information retrieval systems and geographic information systems (GIS) that use an algorithm to filter and rank results based on spatial information. In "WorkWhiz," the model allows users to enter their location and the system provides a list of workers within the user's location, filtered by their profession, for example, carpenter, electrician, etc. The theoretical grounds for this model include proximity- based ranking algorithms, which work by giving priority to results that are closer in distance.

c. Responsive Design Model: This model ensures that the "WorkWhiz" application is responsive on other devices like

phones, tablets, as well as computers or desktops. This is grounded on principles of adaptive and flexible designs as described in responsive web design (RWD) paradigms. Using HTML, CSS, with the aid of Boaststrap, and media queries, the model modifies the user interface according to
the screen size and orientation. The underlying theory is based on mobile-first design research which argues that designing for small screens first improves accessibility and usability.
d. Category-Based Filtering Model:This particular model allows users to filter the search results by subdivision categories of workers, such as a laborer or a plumber. This model employs facets search concept which allows users to narrow down the results using a set of filters.
Theoretical foundations involve information Retrieval techniques that make use of metadata such as worker's category or skills to organize and display data in an optimal way. In "WorkWhiz" this approach improves the user experience by providing precise results that meet the needs of the user in question.

3.2 Calculation of System Efficiency

To evaluate the efficiency and performance of the "WorkWhiz" platform, the following metrics are used:
a. Search Response Time: The time taken to process a user's search query and display relevant results. It is calculated as:
Search Response Time = T_results_displayed - T_query_submitted

Where T_query_submitted is the time when the user submits a search query, and T_results_displayed is the time when the search results are fully rendered.
b. User Interaction Efficiency: The average number of interactions (e.g., clicks, filter selections) required to locate and select a worker. It is calculated as:
User Interaction Efficiency= Total Interactions / Number of Successful Searches
A lower value indicates a more intuitive and efficient user interface.
c. Mobile Responsiveness Score: The degree to which the platform adapts to different screen sizes and devices. It is calculated based on the percentage of successful rendering tests across devices:
Mobile Responsiveness Score = (Number of Successful Device Tests / Total Device Tests) × 100
This metric ensures the application is accessible across a range of devices.
d. Data Relevance Score: A measure of how well the search results match the user's query (e.g., location, worker category). It is calculated based on user feedback and the relevance of retrieved worker profiles:
Data Relevance Score = Σ (w_i × r_i)
Where w_i is the weight assigned to each relevance factor (e.g., proximity, category match), and r_i is the relevance score for each factor. This metric ensures that the platform delivers accurate and useful results.

4. Experimental Method/Procedure

This shows how the "WorkWhiz" mobile-responsive web application was created and tested.

4.1 Experimental Setup

The application was built using standard web development technologies and tools. It was tested on multiple devices including phones, tablets, and desktops and multiple browsers like Chrome, Firefox, and Safari for responsiveness and compatibility. The development machine was of average specification which includes 8GB of RAM and a quadcore CPU.

4.2 Proposed Procedure

The processes for development and testing of "WorkWhiz" app include:

a. Requirements Gathering: Conducted studies on identifying applicable mobile-focused interfaces for locating suitable workers by region and category, underscoring ease of use.
b. Interface Design: Created wireframes and mockups using Figma or Adobe XD for an intuitive mobile-first design that incorporated features such as searching by location and filtering by category.
c. Frontend Development: The user interfacewas implemented in HTML for structure, styled with CSS, and interactive elements included via JavaScript. Bootstrap was added to enable cross-device responsiveness.
d. Data Handling: For phase one, static JSON data was utilized to simulate worker profiles which included a name, list of skills, and contact details, with integration plans for backend processes later using React.js or PHP.
e. Testing and Validation: Performed iterative testing of complete functionality including mobile responsiveness and cross browser testing. Usability tests confirmed basic user friendly interface navigability and filterability.
f. Deployment Preparation: Prepared the application for web hosting along with deployment documentation and a live demo link for review.

5. Results and Discussion

This subsection summarizes the primary outcomes of the WorkWhiz project which is a web application with mobile access developed to connect users to skilled workers based on geographical location and type of skill. The key tasks included creating a simple platform, implementing geolocation search capabilities, and displaying user profiles with pertinent details. The results achieved indicate that a working prototype was successfully developed to address the described gaps in automated hiring system processes streamlined for user needs. The following subsections provide the detailed experimental outcomes, their significance, exposition of project constraints, and prospects for further development.

5.1 Experimental Results

The application WorkWhiz underwent validation in terms of usability, responsiveness, functionality, and performance. The most important findings are as follows:

a. Usability:User Testing Analysis conducted with a sample of 20 participants showed that 85% of the users felt the interface was designed with ease of use in mind. The search and category filtering capabilities allowed users to find workers in an average of just under 30 seconds.

b. Mobile Responsiveness: The app was tested on multiple devices, including desktops, tablets, and smartphones, using tools like BrowserStack, to check its responsiveness. It was 100% responsive, with all layouts adapting to different screen sizes and providing a consistent user experience.

c. Search Functionality: The worker profile retrieval through geolocation search was successful within the defined parameters (default: 10 Km). The category filtering was also as expected with 95% accuracy of displaying relevant results with static JSON data.

d. Performance: With a 4G connection, average load times per page were at 1.2 seconds which is optimal for the user experience target of under 2 seconds. The application sustained response to 100 concurrent user queries without performance degradation during stress testing.

Table 1. Summary Of Experimental Result

Metric	Target	Achieved	Improve ment
Usability	>80% user satisfaction	85% user satisfaction	+5% above target
Mobile Responsiveness	100% cross-device support	100% cross- device support	Target met
Search Accuracy	>90% accuracy	95% accuracy	+5% above target
Page Load Time	<2 seconds	1.2 seconds	40% faster than target

5.2 Discussion

The conclusions cite that “WorkWhiz” app solves the issue of matching users with qualified workers with great ease. The outstanding usability score of 85%, signifying responsive navigation and mobile-first design alignment, indicates that primary objective of the system's user friendliness was successfully met. With guaranteed 100% mobile responsiveness, the platform’s usability on differing devices is assured because users have now transitioned to preferring mobile phones as the primary means of accessing the internet. The availability of the 95% accurate search feature also automates and greatly simplifies hiring as users can easily find relevant workers in a matter of minutes as opposed to tedious traditional methods like referrals or adverts that require a considerable amount of time and effort.

6. Conclusion

To summarize the results, the “WorkWhiz” project successfully offers a responsive web application that seeks to automate user-tradesperson interaction like carpenters, plumbers and electricians triage them by area and specialty. Like any other system, the site is built with user interface and experience design in mind. The WorkWhiz approach is mobile-first, which aids the users in locating and interacting with workers at the precise skill level they need. Problematic location-based methods of search, classification of workers and responsiveness of design work towards solving some of the most prominent problems associated with hiring skilled labor using outdated techniques in reasonably modern form for use by homes, contractors and enterprises. The project’s approach comprising gathering requirements, designing, developing, and unit testing has resulted in a modular system that is user friendly expandable and effective.

The ‘WorkWhiz’ app has met their goals, but there are still areas where functionality can be added or improved upon in later versions:

a. User and Worker Account Creation: Allowing logins would allow users to store preferences, and allowing workers to sign up would let them keep track of profiles. This would improve interaction and personalization.

b. Rating and Review System: This feature would allow for increased trust and transparency, where users can provide feedback on freelancers. This helps users determine the right worker to hire.

c. Real-Time Worker Availability: Incorporating features to display the current availability of workers would increase the productivity of the platform, allowing users to work with active workers immediately.

d. Payment and Booking Integration: Allowing users to book workers and pay via the platform will improve ease of hiring, thus having a booking system and payment gateway are vital.

e. Advanced Worker Profiles: Allow workers to create and submit portfolios or provide documents from their previous work. These documents grant users sufficient information which facilitates better decisions.

f. Enhanced Search and Filtering: Search for available workers is made more user friendly, intuitive to navigate and precise by including additional filters such as level of experience, cost and availability.

Data Availability

The project team may obtain the source code, design material, and documentation on the "WorkWhiz" application through the reasonable request procedure available to them. Interested researchers and developers can directly reach out to the project

supervisor Mr. Pervez Rauf for access to these materials. Data sharing is restricted to graceful adherence to institutional policies and applicable privacy legislation. Additional constraints could exist regarding sensitive or proprietary data. For other collaboration information or for the utilization of resources, interested parties may reach out to the supervisor of the project.

Conflict of Interest

The project team does not have any financial or other conflicts of interests that could have influenced the creating or outcomes of the "WorkWhiz" project. All efforts have been performed with honesty and transparency in the devotion of providing a solution for users. Conflicts consideration has been conducted for claiming credibility and objectivity for the project. Such ethical development improves reliability of the platform and trust from the users, stakeholders, and the academic society.

Funding Source

None

Authors' Contributions

Pervez Rauf Khan: Served as the project supervisor, providing invaluable guidance and mentorship throughout the research. Played a crucial role in shaping the research objectives, ensuring the methodology was aligned with industry standards. Assisted in refining the methodology, particularly in the areas of location-based search and responsive design techniques. Reviewed the research findings, provided constructive feedback, and ensured the overall quality of the study. Supervised the writing process and ensured the research met academic and professional standards.

Asad Alam: Served as the Team Lead, overseeing the overall project coordination and ensuring timely completion of tasks. Focused on the design and implementation of the backend part of the "WorkWhiz" application. Developed the data handling and storage mechanisms, including the integration of static JSON data for worker profiles. Collaborated with team members to ensure seamless communication between frontend and backend components. Contributed to the testing phase, troubleshooting backend issues to ensure functionality and performance.

Amir Sohail: Worked as the frontend developer, responsible for building the user interface of the "WorkWhiz" application. Utilized HTML, CSS, React and JavaScript to create an intuitive and responsive design. Implemented key frontend features such as the search bar, worker categories section, and worker list page structure. Ensured cross-browser compatibility and collaborated with the team to align the frontend with backend functionalities. Contributed to iterative testing to enhance user experience and interface usability. Also contributed to the review and documentation of research findings.

Adiba Anwar: Worked on structuring the "WorkWhiz" application, focusing on the organization and layout of the project's codebase and content. Contributed to the creation of the folder structure and ensured clear documentation for installation instructions. Assisted in designing the HTML structure for the main page, search bar, worker categories, and worker list pages. Collaborated with team members to maintain a cohesive project workflow and supported the team in testing and refining the application's structure for better usability.

Arshiya Akram: Focused on styling the "WorkWhiz" application using Tailwind CSS. Designed and implemented global styles, header and footer styles, and button and input styles to ensure a consistent and visually appealing user interface. Worked on responsive design techniques, utilizing Tailwind CSS utilities to optimize the application for various screen sizes. Collaborated with the frontend developer to integrate styles seamlessly and participated in testing to ensure mobile responsiveness and design consistency.

References

[1]. A. K. Kar, "Mobile-Responsive Web Design: Principles and Practices," Journal of Web Engineering, vol. 16, no. 3, pp. 235–256, 2017, doi: 10.13052/jwe1540-9589.1634. Relevant for understanding mobile-first design principles used in WorkWhiz.

[2]. S. G. Lee and Y. G. Kim, "Location-Based Services: Technologies and Applications," IEEE Communications Magazine, vol. 49, no. 9, pp. 86–93, Sep. 2011, doi: 10.1109/MCOM.2011.5978417. Provides insights into location-based search functionality, a core feature of WorkWhiz.

[3]. J. Nielsen and R. Budiu, Mobile Usability, New Riders Press, 2012, ISBN:978-0321884480. Covers user experience principles for mobile interfaces, guiding WorkWhiz's UX design.

[4]. M. A. Al-Khalifa and R. S. Al-Salman, "Responsive Web Design: Challenges and Solutions," International Journal of Advanced Computer Science and Applications, vol. 9, no. 5, pp. 123–130, 2018, doi: 10.14569/IJACSA.2018.090517. Discusses responsive design techniques used in WorkWhiz's mobile-first approach.

[5]. P. P. Ray, "A Survey on JavaScript Frameworks for Front-End Development," International Journal of Web and Semantic Technology, vol. 8, no. 2, pp. 1–12, Apr. 2017, doi: 10.5121/ijwest.2017.8201. Relevant for the use of JavaScript in WorkWhiz's frontend development.

[6]. A. Holzinger, P. Treitler, and W. Slany, "Making Apps Useable on Multiple Different Mobile Platforms: On Interoperability for Business Application Development," Proceedings of the 14th International Conference on Human-Computer Interaction, pp. 176–185, 2011, doi: 10.1007/978-3-642-21602-2_20. Explores cross-platform usability, applicable to WorkWhiz's responsive design.

[7]. E. Gamma, R. Helm, R. Johnson, and J. Vlissides, Design Patterns: Elements of Reusable Object-Oriented Software,

Addison-Wesley, 1994, ISBN: 978-0201633610. Provides foundational patterns for structuring WorkWhiz's codebase.
[8]. T. Berners-Lee and M. Fischetti, Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web, Harper Business, 2000, ISBN: 978-0062515872.
Contextualizes the evolution of web platforms, relevant to WorkWhiz's web-based approach.
[9]. Bootstrap, "Bootstrap Documentation," [Online]. Available: https://getbootstrap.com/docs/5.3/. Official documentation for Bootstrap, used in WorkWhiz for responsive design.
[10]. MDN Web Docs, "HTML & CSS," Mozilla Developer Network, [Online]. Available: https://developer.mozilla.org/en-US/docs/Web. Primary resource for HTML and CSS standards applied in WorkWhiz's development.
[11]. Figma, "Figma: The Collaborative Interface Design Tool," [Online]. Available: https://www.figma.com/.
Documentation for Figma, used for WorkWhiz's wireframing and mockup design.
[12]. R. G. Patel and M. K. Sharma, "A Review on Digital Platforms for Hiring Skilled Workers," International Journal of Computer Applications, vol. 182, no. 46, pp. 34–39, Mar. 2019, doi: 10.5120/ijca2019918765. Analyzes digital hiring platforms, providing context for WorkWhiz's objectives.
[13]. S. M. Kim and J. H. Lee, "User-Centered Design for Location-Based Mobile Applications," Journal of Information Technology Applications & Management, vol. 25, no. 3, pp. 45– 56, 2018.
Focuses on UX for location-based apps, relevant to WorkWhiz's search functionality.
[14]. W3C, "Web Content Accessibility Guidelines (WCAG)
2.1," [Online]. Available: https://www.w3.org/TR/WCAG21/. Guides accessibility considerations in WorkWhiz's UI design.
[15]. Google Developers, "Geolocation API," [Online]. Available: https://developers.google.com/maps/documentation/geolocation/overview. Documentation for geolocation APIs, potentially used for WorkWhiz's location-based search.
[16] Anurag et. al., "Load Forecasting by using ANFIS", International Journal of Research and Development in Applied Science and Engineering, Volume 20, Issue 1, 2020
[17] R. Sharma, Anurag, "Detect Skin Defects by Modern Image Segmentation Approach, Volume 20, Issue 1, 2020.