ng & Software Engineering

CSE5015 Computing Project Proposal Approval Sheet

Student Use only;

Group Name		Attempt No	1
Title of the project	GIG CONNECT		
System platform	☒ Web Application ☐ Desktop Application ☐ Mobile Application ……		
Technology	HTML, CSS, JavaScript, PHP, Wamp Server		
Programming methodology	☐ Structured System Analysis and Design Methodology Object Oriented Analysis and Design Methodology		
SDLC	Agile Methodology		

Student No	112/	Full Name	Aleem Shakeer
Roles & Responsibilities. **Project Manager:** Coordination of the team, Planning, and risk management.			
Student No	112/	Full Name	Kavindu Nethmal
Roles & Responsibilities. **System Analyst:** Requirement Gathering Analyzing, Process Optimization			
Student No	112/144	Full Name	Emethra Nethmini
Roles & Responsibilities. **UI/UX Engineer:** User researching, Interface Designing			
Student No	112/52	**Full Name**	Sadeesha Heshan
Roles & Responsibilities. **Database Engineer:** Database Designing, Data Management, Performance Optimization			
Student No	112/57	Full Name	Gavindu Prathap
Software Engineer: Software developments, Code Writing, Debugging the code			

Contents

TABLE OF FIGURES

TABLE OF TABLES

GIG CONNECT

Connecting People with Independent Contractors and Gig Workers

Project Overview

Objective:
To create a seamless, efficient platform that connects customers with independent contractors and gig workers, facilitating transparent interactions and reliable service provision.

Key Features:

- **User Accounts:** Admins, customers, and gig workers.
- **Customer Functionality:** Post job requests, receive bids, review and rate contractors.
- **Gig Worker Functionality:** Browse job postings, propose prices, negotiate, and build reputation through reviews.
- **Interaction and Feedback:** Smooth communication, price negotiations, and a feedback system to ensure trust and quality.

Introduction

Welcome to **GigConnect**, a pioneering app designed to streamline the process of connecting customers with independent contractors and gig workers. In a world where convenience and efficiency are paramount, GigConnect emerges as a solution to bridge the gap between those in need of skilled assistance and those offering specialized services.

Purpose of GigConnect

GigConnect aims to transform how individuals find and hire independent contractors for various tasks. By leveraging technology, our platform facilitates a smooth, transparent, and efficient connection process. Whether you are a homeowner needing a quick repair or a contractor seeking new opportunities, GigConnect is here to simplify and enhance your experience.

How It Works

1. **User Accounts:**
    - **Admins:** Oversee the platform's operations and ensure smooth functionality.
    - **Customers:** Create and manage job requests, receive and evaluate offers, and provide feedback.
    - **Gig Workers (Contractors):** Browse job postings, propose pricing, negotiate terms, and build their reputation through customer reviews.
2. **Customer Side:**
    - **Account Creation and Login:** Easily set up an account and log in to access services.

- **Job Posting:** Select a relevant category, describe the problem, upload images, and post the request.
- **Offer Evaluation:** View bids from contractors, negotiate, and choose the best offer.
- **Feedback:** Rate and review completed jobs to aid future customers in their decision-making process.

3. **Gig Worker/Contractor Side:**
   - **Account Creation and Login:** Set up an account to start engaging with job opportunities.
   - **Job Browsing:** Access a list of posted jobs relevant to your expertise.
   - **Proposals and Negotiation:** Offer prices, negotiate terms with customers, and finalize agreements.
   - **Reputation Building:** Accumulate positive reviews to enhance your profile and attract more clients.

4. **Interaction and Feedback:**
   - **Seamless Communication:** Facilitate effective discussions between customers and contractors.
   - **Transparent Pricing:** Ensure clear and fair pricing through direct negotiations.
   - **Reputation System:** Utilize customer ratings and reviews to build credibility and trust.

Our Vision

GigConnect is designed to create a reliable, transparent, and user-friendly environment where customers and contractors can connect efficiently. Our goal is to empower both parties by providing a platform that enhances communication, fosters trust, and supports high-quality service delivery. By bridging the gap between demand and supply in the gig economy, GigConnect not only simplifies the hiring process but also promotes transparency and accountability.

We are excited to embark on this journey and invite you to experience the future of gig and contractor services with GigConnect.

Background Study

The development of the GigConnect app focuses on creating a seamless platform for connecting customers with independent contractors and gig workers. This project addresses the challenges both customers and contractors face in finding reliable service providers and job opportunities, respectively. On the customer side, the app facilitates problem identification, service request posting, contractor engagement, and review/rating functionalities. For contractors, it offers a space to browse job opportunities, propose service pricing, and negotiate with customers, enhancing transparency and communication.

Conducting a thorough background study is crucial for understanding the dynamics of the gig economy and how technological advancements can streamline the process. The research involves an in-depth analysis of the workflows associated with job posting, contractor

selection, and job completion. Identifying the pain points and potential enhancements through technology helps ensure that the system effectively caters to both customers and contractors.

Engaging with end users: customers and contractors through surveys and interviews will yield valuable insights into their expectations and challenges. Understanding the needs and preferences of these users is critical for designing a user-friendly platform that fosters trust and ease of use, thus ensuring GigConnect becomes a competitive player in the gig service marketplace.

The study also involves reviewing current regulations around online contracting and gig work, particularly in relation to data protection, privacy, and payment security. Since the platform handles sensitive information such as user details, problem descriptions, and payment information, ensuring compliance with data protection legislation and implementing robust security measures like encryption and secure authentication protocols are paramount.

Additionally, the analysis of existing marketplace platforms and contractor services provides an understanding of the competition, highlighting the advantages and limitations of similar systems. This insight informs GigConnect's unique value proposition, allowing for the differentiation of the platform through features such as direct communication, user ratings, and the streamlined contractor-customer matching process.

Scalability and future development are also key factors in the background study. With the growing demand for gig work across various industries, it's important to design a system that can adapt to the changing needs of customers and contractors. This ensures the app remains relevant and can evolve alongside technological advancements. Prioritizing usability and user experience (UX), the interface will be simple and intuitive, allowing users to easily navigate the app's features, thereby increasing satisfaction and long-term engagement.

In summary, the background study for GigConnect emphasizes a comprehensive understanding of user needs, regulatory compliance, marketplace dynamics, and future scalability. By focusing on user-centric design and security, the platform will effectively bridge the gap between customers and contractors, providing a smooth and efficient service experience.

Objectives for GigConnect

1. **Analyzing Existing Platforms:** Conduct a comprehensive evaluation of current gig work platforms and contractor services, identifying key areas for improvement. This will help optimize GigConnect for a more seamless user experience and higher efficiency in connecting customers with contractors.
2. **Understanding Market Dynamics:** Investigate the latest trends in the gig economy and independent contractor services. By examining how technological advancements and market behavior impact both customers and contractors, GigConnect will be developed to align with current demands and expectations.
3. **Researching User Requirements:** Perform in-depth research on customer and contractor needs, expectations, and challenges. By focusing on user feedback, GigConnect will tailor its features to enhance the experience for both parties, addressing pain points and fostering positive interactions.
4. **Evaluating Industry Trends:** Assess broader technological and industry shifts, including the rise of gig work and on-demand services. This ensures that GigConnect

remains relevant and adaptable in an evolving marketplace, positioning it for long-term success.

5. **Establishing a Strong System Framework:** Synthesize insights from market trends, user research, and competitive analysis to develop a robust framework for GigConnect. This framework will support efficient operations, scalability, and continuous improvement, ensuring that the app effectively meets user needs and stays ahead of industry changes.

Existing Gig Work Platforms:

Creating an app like GigConnect is a complex project that requires careful planning. The first step involves conducting thorough research on existing gig work platforms that connect customers with independent contractors. By comparing and analyzing these platforms, we can identify the similarities, key features, and pain points that they address. This research will allow us to incorporate all the essential features into GigConnect, while also adding unique functionalities that make it stand out in the market.

This evaluation phase is critical for informing the design and development stages of the project. It enables us to make data-driven decisions that ensure GigConnect meets user requirements and exceeds customer expectations. By understanding the strengths and weaknesses of current platforms, we can implement a solution that provides seamless communication, efficient job matching, and a transparent experience for both customers and gig workers. This approach ultimately contributes to delivering a superior product that aligns with the needs of the gig economy.

1. Upwork Website

Desktop View

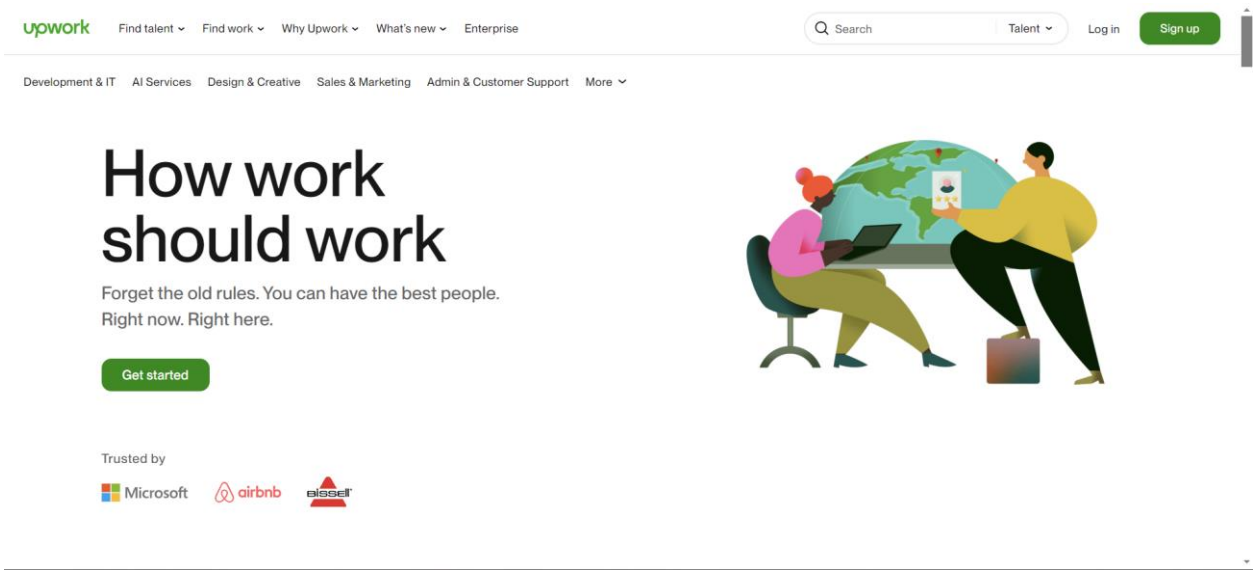


Figure 1 Upwork Desktop View

Mobile View

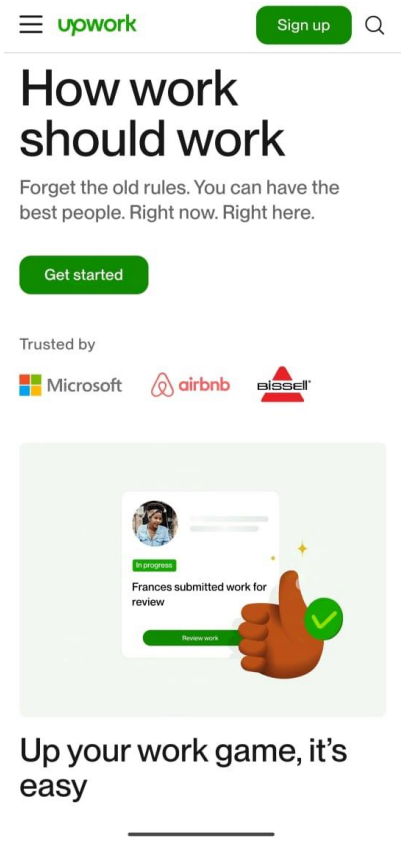


Figure 2 Upwork mobile view

I. User Experience Design (UX)

Upwork places a high priority on user experience, ensuring seamless navigation throughout the platform. The homepage presents a clean and modern design, with a clear call to action for both freelancers and clients. Key features, such as searching for talent and posting jobs, are made accessible via a well-structured navigation bar. Additionally, Upwork's consistent use of color and minimalistic design enhances the professionalism and ease of use. Visual cues and subtle animations guide users, making the platform intuitive and user-friendly for various types of projects.

II. Responsive Web Design (RWD)
Upwork's design is fully responsive, allowing freelancers and clients to manage projects efficiently from any device. Whether accessing the platform on a desktop, tablet, or smartphone, the layout dynamically adjusts to fit the screen size, ensuring the same high-quality experience across all platforms. Images, text, and functional elements such as buttons and search bars resize fluidly while maintaining optimal clarity and usability. This ensures that both freelancers and clients can connect, collaborate, and manage tasks seamlessly regardless of their device.

III. Security
Security is a crucial aspect of Upwork's platform, especially with the exchange of sensitive information such as payments and personal data. The platform implements robust authentication processes, including two-factor authentication, to ensure that accounts remain secure. User profiles and financial transactions are protected through encrypted communications, providing peace of mind for both freelancers and clients. Upwork also enforces strict verification processes, ensuring that only legitimate users gain access to the platform and sensitive data.

Front End

1. **HTML and CSS** are used to structure and style the interface, ensuring a consistent and visually appealing user experience.
2. **JavaScript** enhances interactivity, enabling dynamic content such as live project updates, real-time messaging, and notifications.
3. **Responsive design** ensures the site adapts to different screen sizes, delivering a smooth experience across devices.

Back End

1. **Database Management:** MySQL and other database solutions are used to store user data, project details, and transactions.
2. **Session Management** ensures secure and continuous user sessions, preserving account data across pages and interactions.
3. **Payment Gateway Integration:** Secure handling of financial transactions between freelancers and clients, with options for various payment methods globally.
2. Uber Website

Desktop View

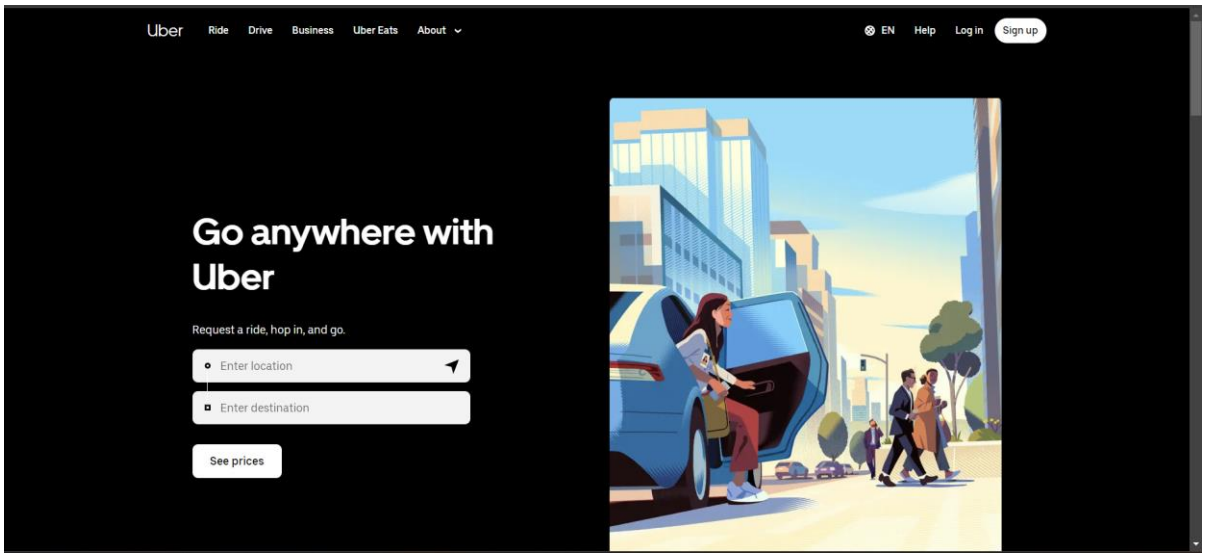


Figure 3 uber desktop view

Mobile View

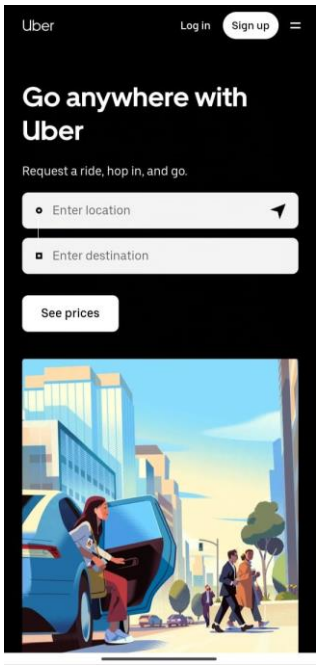


Figure 4 uber mobile view

I. User Experience Design (UX)

Uber prioritizes a streamlined and user-centric experience, making it effortless for users to navigate the platform. The homepage features an intuitive design with a prominent, user-friendly interface. Clear calls to action for both riders and drivers are strategically placed, facilitating easy access to ride requests, driver sign-ups, and account management. The use of vibrant, high-contrast colors and engaging visual elements enhances user engagement while maintaining simplicity. Interactive elements, such as real-time ride tracking and dynamic pricing updates, are seamlessly integrated to provide an engaging and efficient user experience.

II. Responsive Web Design (RWD)

Uber's website is designed to be fully responsive, offering a consistent and optimized experience across all devices. The layout adjusts fluidly to fit various screen sizes, including desktops, tablets, and smartphones. Key features, such as booking rides and managing driver profiles, are accessible and function seamlessly on any device. Images and content dynamically resize, ensuring clarity and usability regardless of the user's device. This responsive approach guarantees that users have a smooth and effective interaction with the platform, whether they are on the go or at their desktop.

III. Security

Uber implements comprehensive security measures to protect user data and financial transactions. The platform utilizes advanced encryption protocols to safeguard personal and payment information, ensuring secure communication between users and the platform. Multi-factor authentication is employed to enhance account security, reducing the risk of

unauthorized access. Regular security audits and updates help maintain a secure environment, while Uber's strict verification procedures ensure that only legitimate users and drivers are active on the platform.

Front End

1. **HTML and CSS**: These technologies are used to structure and style the website, creating a visually appealing and consistent user interface.
2. **JavaScript**: Enhances interactivity and functionality, such as real-time ride tracking, dynamic fare calculations, and user notifications.
3. **Responsive Design**: Ensures that the website adapts to various screen sizes and devices, providing a smooth experience whether on mobile, tablet, or desktop.

Back End

1. **Database Management**: Uber uses robust database systems like PostgreSQL and other scalable solutions to manage user data, ride details, and transaction records.
2. **Session Management**: Secure handling of user sessions ensures that ride requests and driver statuses are accurately maintained across interactions.
3. **Payment Gateway Integration**: Facilitates secure and efficient processing of financial transactions, supporting various payment methods and currencies globally.

3. TaskRabbit website

Desktop View

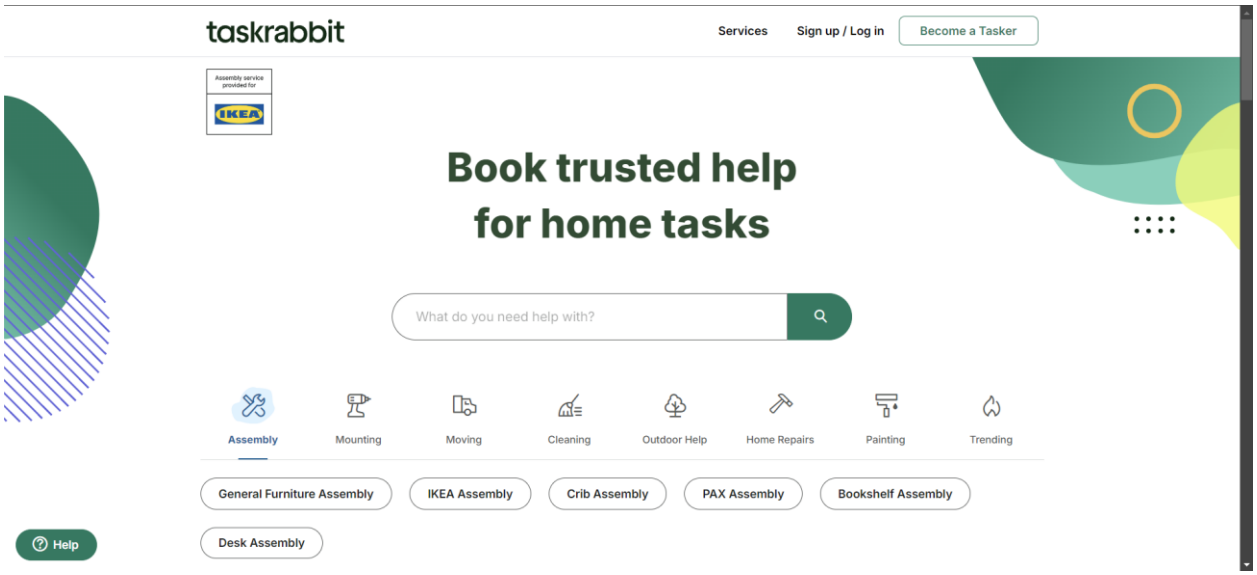


Figure 5 TaskRabbit desktop view

Mobile View

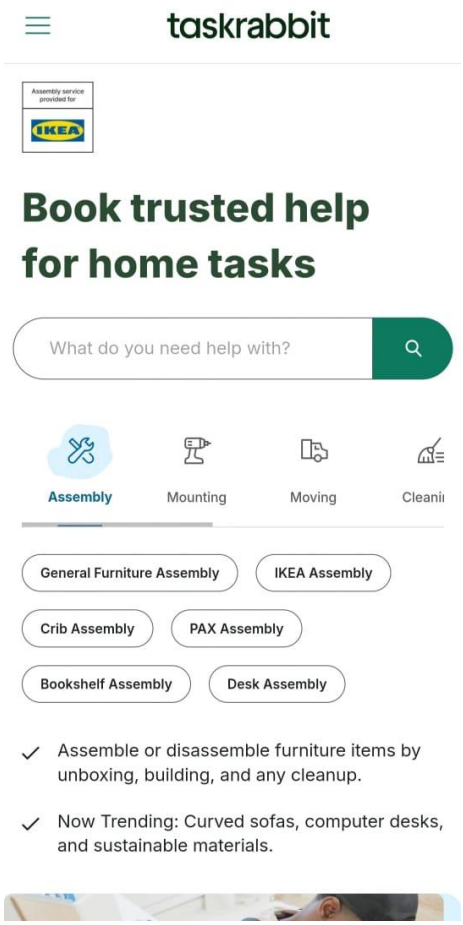


Figure 6 TaskRabbit mobile view

i. User Experience Design (UX)

TaskRabbit emphasizes a straightforward and user-focused experience, designed to connect users with local taskers efficiently. The homepage is clean and well-organized, with an

intuitive layout that highlights key functions such as posting tasks, finding taskers, and managing account settings. Prominent calls to action are strategically placed to guide users through task requests, tasker searches, and booking processes. The website utilizes a cohesive color scheme and clear typography to enhance readability and engagement. Interactive features, such as real-time task tracking and in-app messaging, are seamlessly integrated to ensure a smooth and effective user experience.

ii. Responsive Web Design (RWD)

TaskRabbit's website is built with responsiveness in mind, ensuring that users have a consistent and optimized experience across different devices. The layout adjusts fluidly to various screen sizes, including desktops, tablets, and smartphones. Essential features, such as task posting and tasker profiles, are easily accessible and functional on any device. Images, text, and interactive elements resize dynamically, maintaining clarity and usability regardless of the screen size. This responsive design approach ensures that users can efficiently navigate and interact with the platform whether they are using a mobile device or a desktop computer.

iii. Security

TaskRabbit prioritizes security to protect user data and transactions. The platform employs robust encryption protocols to safeguard personal information and financial transactions, ensuring secure communication between users and taskers. Multi-factor authentication is used to enhance account security, mitigating the risk of unauthorized access. Regular security assessments and updates help maintain a secure environment, while TaskRabbit's verification processes ensure that only verified taskers and users are active on the platform.

Front End

1. **HTML and CSS**: These technologies are employed to structure and style the website, providing a visually appealing and user-friendly interface.
2. **JavaScript**: Enhances interactivity with features such as dynamic task updates, real-time notifications, and interactive search functions.
3. **Responsive Design**: Ensures that the website adapts seamlessly to different devices and screen sizes, delivering a smooth and consistent experience across mobile, tablet, and desktop platforms.

Back End

1. **Database Management**: TaskRabbit uses scalable database systems like PostgreSQL or MongoDB to manage user profiles, task details, and transaction records.
2. **Session Management**: Secure handling of user sessions ensures that task requests, user profiles, and tasker interactions are accurately maintained across the platform.
3. **Payment Gateway Integration**: Facilitates secure processing of payments and financial transactions, supporting various payment methods and currencies to cater to a global user base.

Problem Statement

In today's fast-paced world, finding reliable, skilled workers for specific, short-term tasks can be a significant challenge for many individuals and businesses. Whether it's a homeowner

needing a quick plumbing fix or a small business requiring an expert for a one-time project, the process of locating, vetting, and hiring independent contractors or gig workers is often time-consuming, inefficient, and fraught with uncertainty.

Current Challenges

1. **Lack of Centralized Platforms:** While some platforms exist for connecting gig workers with potential clients, many are either too broad in scope, not user-friendly, or lack the specific focus needed for certain trades and services. This fragmentation leaves customers and contractors alike struggling to find the right match.
2. **Trust and Accountability Issues:** Customers often face difficulty in assessing the reliability and quality of a contractor's work before hiring them. Existing platforms may provide some information, but many still rely heavily on customer reviews, which can be inconsistent or manipulated. This makes it challenging for customers to make informed decisions and for contractors to build a reputable online presence.
3. **Communication Barriers:** Effective communication between customers and contractors is vital to ensure the successful completion of a job. However, many existing systems do not facilitate direct, real-time communication, leading to misunderstandings, delays, and unsatisfactory outcomes.
4. **Inefficient Hiring Process:** The current process for hiring independent contractors typically involves multiple steps, including searching, vetting, contacting, negotiating, and finally, hiring. Each step can be cumbersome and time-consuming, especially for customers who require quick solutions to urgent problems.

Specific Area / Business Process Identified:

The specific area where this problem is most pronounced is in the home services sector, where homeowners regularly need skilled workers for tasks such as plumbing, electrical work, painting, and general repairs. The inefficiency in connecting homeowners with reliable contractors results in wasted time, increased costs, and often, unsatisfactory work quality.

Conclusion

The problem of connecting customers with reliable gig workers in the home services sector is clear and pressing. GigConnect aims to solve this by providing a centralized, transparent, and efficient platform that bridges the gap between customers' needs and contractors' skills. Through enhanced communication, accountability, and a streamlined hiring process, GigConnect will address the inefficiencies and trust issues that currently plague this area of the gig economy.

Literature Review

The gig economy has seen exponential growth in recent years, driven by the increasing demand for flexible work arrangements and the rise of digital platforms that connect independent contractors with potential clients. As GigConnect aims to create a seamless and efficient connection between customers and skilled workers, it is essential to examine

existing systems and technologies that serve similar purposes. This literature review will analyze several prominent platforms in the gig economy, assess their strengths and weaknesses, and identify opportunities for improvement that GigConnect seeks to address.

1. Upwork

Upwork is one of the largest freelancing platforms globally, connecting clients with a wide range of professionals, from writers and designers to software developers and consultants.

Strengths:

- **Comprehensive Marketplace:** Upwork offers a vast array of services, making it a go-to platform for clients with diverse needs.
- **Escrow System:** The platform provides an escrow payment system that ensures funds are securely held until the client approves the work, offering protection for both parties.
- **Rating and Review System:** Clients can leave feedback and ratings for freelancers, which helps build trust and credibility.

Weaknesses:

- **Broad Focus:** Upwork's broad focus on various professional services can make it challenging for clients to find specialized contractors for specific tasks, such as home repairs or manual labor.
- **Complex Fee Structure:** The platform's fee structure, which includes service fees and membership plans, can be confusing and costly for both clients and freelancers.
- **Overwhelming Competition:** With a large number of freelancers on the platform, it can be difficult for new or niche contractors to stand out, leading to a "race to the bottom" in terms of pricing.

2. TaskRabbit

TaskRabbit is a platform specifically designed for connecting users with local service providers for everyday tasks, including cleaning, moving, and home repairs.

Strengths

- **Local Focus:** TaskRabbit emphasizes connecting users with local contractors, which is beneficial for tasks that require physical presence, such as home repairs.
- **Tasker Verification:** The platform has a robust verification process for its "Taskers," which includes background checks, ensuring a higher level of trust and safety.
- **Ease of Use:** TaskRabbit's user interface is simple and intuitive, making it easy for customers to post tasks and for contractors to find work.

Weaknesses

- **Limited Scope:** While TaskRabbit excels in providing services for everyday tasks, its scope is limited compared to platforms like Upwork, making it less versatile for clients with specialized or niche needs.
- **Pricing Transparency:** The platform allows Taskers to set their rates, which can lead to significant price variation and sometimes a lack of transparency for customers.
- **Availability:** TaskRabbit is only available in certain cities, limiting its reach and utility for users in non-metropolitan areas.

3. Thumbtack

Thumbtack is another platform focused on connecting customers with professionals in various service categories, including home services, events, wellness, and more.

Strengths

- **Diverse Service Offerings:** Thumbtack offers a wide range of services, from home repairs to personal training, making it a versatile platform for users with various needs.
- **Custom Quotes:** The platform allows service providers to send custom quotes to customers, which can lead to more tailored and competitive pricing.
- **Customer Reviews:** Thumbtack's review system helps customers make informed decisions by providing insights into the experiences of past clients.

Weaknesses

- **Lead Pricing:** Thumbtack charges service providers for leads, which can become expensive, especially if the leads do not convert into actual jobs.

- **User Experience:** Some users have reported that the platform can be overwhelming due to the sheer number of service providers and quotes, making it difficult to choose the right contractor.
- **Limited Interaction:** Thumbtack's communication tools are not as robust as those of other platforms, potentially leading to misunderstandings or miscommunication between clients and contractors.

4. Fiverr

Fiverr is a global online marketplace that offers services starting at $5. The platform is popular for its wide range of digital services, including graphic design, writing, and programming.

Strengths

- **Low Entry Barrier:** Fiverr's $5 starting price point makes it accessible to clients with smaller budgets and allows freelancers to offer entry-level services.
- **Wide Range of Services:** The platform supports a broad spectrum of services, catering to both digital and creative industries.
- **Gig Packages:** Fiverr allows sellers to offer gig packages at different price points, giving customers options based on their budget and needs.

Weaknesses

- **Quality Variability:** Due to the low starting prices, there is often significant variability in the quality of services provided, which can be a risk for clients seeking high-quality work.
- **Overemphasis on Price:** The platform's emphasis on low-cost services can lead to undervaluation of skilled work, affecting both the quality of services and the livelihood of freelancers.
- **Limited Communication:** Fiverr's communication tools are limited, which can hinder effective collaboration on more complex projects.

5. GigConnect's Unique Value Proposition

While the platforms mentioned above have made significant strides in the gig economy, GigConnect seeks to address several gaps in the market:

- **Specialization in Home Services:** Unlike broader platforms like Upwork and Fiverr, GigConnect will focus specifically on connecting customers with skilled contractors

for home services. This specialization will enable the platform to provide more tailored features and a better user experience for both customers and contractors.

- **Enhanced Communication Tools:** GigConnect will prioritize real-time, direct communication between customers and contractors, allowing for clearer discussions about project requirements, pricing, and timelines. This will help mitigate the misunderstandings and delays that often occur on other platforms.
- **Transparent Pricing and Trust:** GigConnect will incorporate a transparent pricing system with clear guidelines, ensuring that both customers and contractors understand the costs involved. Additionally, the platform will implement robust verification and rating systems to build trust and accountability.
- **Local Focus with Broad Availability:** While focusing on local services like TaskRabbit, GigConnect aims to expand its availability beyond metropolitan areas, making it accessible to a wider range of users.

In conclusion, GigConnect's development is informed by the strengths and weaknesses of existing gig economy platforms. By addressing specific gaps, particularly in the home services sector, GigConnect aims to create a more efficient, trustworthy, and user-friendly solution that better serves both customers and independent contractors.

Proposed Solution

Overview of the Solution

GigConnect is designed to streamline the connection between customers and independent contractors. The system facilitates seamless interactions, including job postings, bidding, negotiations, and feedback collection. This section outlines the system's architecture, implementation plan, and technological framework.

Table 1 user interactions

Component	Description	Interaction
Customer	Can create an account, post job requests, view offers, and rate contractors.	Uses the system to interact with contractors.
Contractor	Can create an account, browse jobs, submit bids, and negotiate.	Uses the system to find and apply for jobs.
System	Manages user accounts, job postings, bids, reviews, and communication.	Facilitates interaction between customers and contractors.

System Architectures

The architecture of GigConnect is designed to ensure robust and scalable functionality. It integrates various components to support customer and contractor interactions efficiently.

Table 2 System Architectures

Component	Description	Interaction
Client	User interface for accessing the application.	Sends requests to the server.
Server	Handles application logic and processes requests.	Communicates with the database.
Database	Stores user data, job posts, bids, and reviews.	Provides data to the server.

Table: Component Overview Table

Table 3 Component Overview Table

Component	Description
User Management	Handles user registration, login, and profile management.
Job Posting	Allows customers to post job requests and upload relevant details.
Bid Management	Enables contractors to view job posts, submit bids, and negotiate.
Review System	Collects and displays customer feedback and ratings for contractors.
Communication	Facilitates direct communication between customers and contractors.

Implementation Plan

The implementation plan outlines the phases, milestones, and deliverables necessary to develop GigConnect.

Table: Development Phases Table

Table 4 Implementation Plan

Phase	Milestone	Deliverables
Planning	Requirement Gathering	Project plan, requirements document
Design	System Design	Design documents, architecture diagrams
Development	Coding and Implementation	Functional system modules, codebase

Testing	System Testing	Test cases, bug reports
Deployment	Launch and Rollout	Live system, user manuals
Maintenance	Ongoing Support and Updates	Regular updates, support documentation

Integration and Compatibility

GigConnect is designed to integrate seamlessly with existing systems and technologies.

Diagram: Integration Diagram

- *Description:* The Integration Diagram illustrates how GigConnect integrates with other systems, such as payment gateways, email services, and notification systems. This diagram shows the points of interaction and data exchange between GigConnect and external services.

Technology Stack

GigConnect utilizes a specific technology stack to support its functionalities.

Table: Technology Stack Table

Table 5 Technology Stack Table

Technology	Purpose
HTML	Structure and layout of web pages
CSS	Styling and visual presentation
PHP	Server-side scripting and logic
MySQL	Database management and storage

Justification to the Solution

Addressing the Identified Problem

GigConnect directly addresses the need for a transparent, efficient, and user-friendly platform to connect customers with contractors.

Table: Problem-Solution Mapping Table

Table 6 Problem-Solution Mapping Table

Identified Problem	Proposed Solution
Inefficient job matching	GigConnect's job posting and bidding system ensures that customers find suitable contractors quickly.
Lack of transparency in negotiations	The platform enables direct communication and detailed bids, improving transparency.
Difficulty in assessing contractor quality	Customer reviews and ratings help in evaluating contractors' performance.

Benefits and Advantages

The proposed solution offers several advantages over existing systems.

Table: Benefits Comparison Table

Table 7 Benefits Comparison Table

Feature	GigConnect	Existing Solutions
Job Posting and Bidding	Streamlined and efficient	Often complex and less user-friendly
Contractor Reviews	Integrated and accessible	May require external tools or sites
Communication Tools	Built-in messaging and negotiation	Often fragmented or lacking features

Evidence and Support

The effectiveness of GigConnect is supported by various data and evidence.

Diagram: Evidence Diagram

- *Description:* The Evidence Diagram visually represents data supporting the effectiveness of GigConnect, such as user engagement statistics, satisfaction rates, and operational metrics.

Table: Case Study Comparison Table

Table 8 Case Study Comparison Table

Case Study	Description	Relevance to GigConnect
Case Study 1	Overview of a similar platform's success	Provides insights into potential benefits and challenges.
Case Study 2	Analysis of customer feedback on existing solutions	Highlights areas where GigConnect can offer improvements.

Feasibility

GigConnect is feasible considering the practical aspects of implementation and alignment with project goals.

Table: Feasibility Table

Table 9 Feasibility Table

Aspect	Considerations
Technical Feasibility	Utilizes established technologies (HTML, CSS, PHP, MySQL).
Cost Feasibility	Development and maintenance costs are manageable.
Resource Availability	Required skills and tools are readily available.
Alignment with Goals	Meets objectives of improving job matching and contractor transparency.

Scope

“GigConnect” is an application designed to bridge the gap between customers seeking services and independent contractors or gig workers offering those services. The system will provide a platform where customers can easily find and hire skilled workers for various tasks while enabling gig workers to find jobs that match their expertise. The scope of this project includes the development, implementation, and maintenance of the "GigConnect" app, ensuring a seamless experience for all users.

Deliverables

- A fully functional "GigConnect" Web page with a responsive design.
- Secure login and user management for admins, customers, and gig workers.
- Integrated messaging system for seamless communication.
- Rating and review system to build trust and reputation.
- Admin dashboard for system management and monitoring.

Functional Requirements

Customer, Admin, Gig Workers-Login and Register

Customer :-

- **Login:**
  - Customers should securely log in using an email/username and password.
- **Register:**
  - New customers should be able to create an account by providing necessary information such as name, email, contact details, and password creation.
- **Profile Management:**
  - Customers should be able to edit their profiles, update contact details, and manage preferences.
- **Search and Browse Gig workers:**
  - Customers should be able to search for gig workers by skill, location, availability, and ratings.
- **Job Posting:**
  - Customers should be able to post jobs with detailed descriptions, budgets, and deadlines.
- **Gig workers Selection:**
  - Customers should be able to review gig workers profiles, ratings, and previous work, and select the most suitable gig worker for their job.
- **Payment:**
  - Customers should be able to securely make payments for services through the app.
- **Job History and Management:**

  - Customers should have access to their job history, view ongoing projects, and manage their current contracts.

- **Ratings and Reviews:**
  - Customers should be able to rate and review gig workers based on their service quality.

- **Notifications:**
  - Customers should receive notifications for job updates, payment confirmations, and other relevant information.

Admin :-

- **Admin Login:**
  - Admins should securely log in using a username and password.
- **User Management:**
  - Admins should be able to manage both customer and gig workers accounts, including activation, deactivation, and role assignments.
- **Job and Gig workers Monitoring:**
  - Admins should have oversight of all job postings, gig workers activities, and customer interactions for quality control and compliance purposes.
- **Reporting and Analytics:**
  - Admins should access various reports related to platform performance, user activities, financial transactions, and other key metrics.
- **Content Management:**
  - Admins should manage the content displayed on the platform, including updates to FAQs, terms of service, and promotional materials.
- **Dispute Resolution:**
  - Admins should have tools to manage disputes between customers and gig workers, including mediation and enforcement of platform policies.
- **Notifications:**
  - Admins should receive notifications for critical platform events such as system errors, financial discrepancies, or user complaints.

Gig workers :-

- **Login:**
  - Gig workers should securely log in using an email/username and password.
- **Register:**
  - New gig workers should be able to create an account by providing necessary information such as name, email, contact details, skill sets, and uploading credentials or portfolios.
- **Profile Management:**
  - Gig workers should be able to manage their profiles, update skills, availability, and contact details, and upload work samples or certifications.
- **Job Browsing and Application:**
  - Gig workers should browse available jobs by category, location, and payment, and apply for jobs that match their skills.

- **Job Management:**
  - Gig workers should manage accepted jobs, communicate with customers, and update job progress.
- **Payment Management:**
  - Gig workers should track their earnings, manage payment methods, and withdraw funds securely.
- **Ratings and Reviews:**
  - Gig workers should be able to view customer feedback, respond to reviews, and improve their service based on ratings.
- **Notifications:**
  - Gig workers should receive notifications for job opportunities, job acceptance, payment updates, and other relevant events.XS

Search Functionality:

- **Search for Gig workers:**
  - Users (customers) should be able to search for Gig workers based on various criteria such as skills, location, name, ratings, and availability.
- **Filtering and Sorting:**
  - The search functionality should support filtering by Gig workers skills, experience level, pricing, and customer ratings.
  - Sorting options should be available to help users order results by relevance, price, or ratings.

Gig workers Profile View:

- **Profile Details:**
  - Users should be able to view detailed profiles of Gig workers, including their name, photo, skill set, experience, location, and contact information.
- **Certifications and Awards:**
  - The profile should display any certifications, awards, or recognitions received by the Gig workers, along with a brief description of their significance.
- **Portfolio:**
  - Gig workers should be able to showcase their portfolio, including images, videos, or links to previous work.

Job Posting and Application Integration:

- **Job Posting from Profile:**
  - Users should have the option to directly post a job or invite a contractor to apply for a job from the gig worker's profile view.
- **Availability Display:**
  - The profile should display the gig worker's availability for new jobs, allowing users to select a suitable time frame for the project.

Customer Reviews and Ratings:

- **View Reviews:**

    - Users should be able to view customer reviews and ratings for the gig workers. The profile should display an average rating and allow users to read individual reviews.
- **Leave a Review:**
    - After completing a job, customers should be able to leave a review and rating for the gig workers.

Specialization Details:

- **Skill Details:**
    - Users should be able to view details about the gig worker's specific areas of expertise, certifications, and tools or technologies they are proficient in.

Job Management:

- **Job Listing:**
    - Customers and gig workers should be able to view a list of their ongoing, completed, and pending jobs. The list should display relevant details such as job title, status, gig workers/customer name, and agreed-upon price.
- **Job Details View:**
    - Users should be able to click on a job to view more detailed information, including job description, agreed-upon price, timelines, and any communication history between the customer and gig workers.

Payment Information:

- **Payment Status:**
    - The job detail's view should display the payment status, including any outstanding balances, payments made, and payment due dates.
- **Secure Payments:**
    - Customers should be able to securely make payments to contractors through the platform, with payment statuses updated in real-time.

Job Status Tracking:

- **Status Updates:**
    - Users should be able to see the status of the job, such as "in progress," "completed," or "canceled." If the job is pending or canceled, the details view should provide information about the reason for the status change.
- **Milestone Tracking:**
    - For larger jobs, the platform should allow for the tracking of milestones, with each milestone having a status update and related payment information.

Job Notes and Communication:

- **Job Notes:**
    - Users should have access to any notes or instructions related to the job, such as project requirements, deliverables, and any additional comments or requests.

- **Messaging System:**
  - The platform should implement a messaging system that allows customers and gig workers to communicate directly within the platform. This system should support text, attachments, and real-time chat.

Cancellation/Rescheduling Options:

- **Cancellation:**
  - Users should be able to cancel jobs directly from the job detail's view, with the platform handling any related payment reversals or fees.
- **Rescheduling:**
  - If applicable, users should be able to reschedule jobs, with the platform notifying the other party and updating the job's timeline accordingly.

Customer Support:

- **Support Channels:**
  - The platform should offer multiple support channels for users, including email, live chat, and phone support. These should be accessible from within the app for quick resolution of issues.
- **Help Center:**
  - Users should have access to a help center with FAQs, tutorials, and contact options for more detailed support.

Real-Time Notifications:

- **Job Updates:**
  - Users should receive real-time notifications for job status updates, messages from the other party, payment confirmations, and other relevant events.
- **Reminders:**
  - The platform should send reminders for upcoming milestones, pending payments, and deadlines.

Non-Functional Requirement:

1. Security:
   - Data Encryption:
     - All sensitive data, including user information and communications, will be encrypted.
   - Authentication:
     - Multi-factor authentication will be implemented to ensure secure login.
   - Role-based Access Control:
     - Different levels of access will be assigned to customers, contractors, and admins.
2. Performance:
   - Scalability:
     - The system will be designed to handle a growing number of users and jobs without performance degradation.
   - Response Time:
     - The app will respond to user actions within 2-3 seconds for a smooth experience.

- Availability:
  - The system will maintain 99.9% uptime, always ensuring availability.

3. Usability:
   - User Interface:
     - The app will have an intuitive, user-friendly interface with responsive design, accessible across various devices (mobile, tablet, desktop).
   - Accessibility:
     - The app will follow accessibility standards to ensure it is usable by individuals with disabilities.
4. Maintainability:
   - Modular Design:
     - The system will be developed using a modular architecture to facilitate easy updates and maintenance.
   - Error Logging:
     - The system will include comprehensive error logging and monitoring to quickly identify and resolve issues.
5. Privacy:
   - Data Privacy:
     - User data will be handled according to privacy laws and best practices, with clear policies for data usage and retention.
   - User Control:
     - Users will have control over their data, with options to update or delete their information.
6. Reliability:
   - Backup and Recovery:
     - Regular data backups will be performed, and a disaster recovery plan will be in place to ensure data integrity in case of system failures.
   - Fault Tolerance:
     - The system will be designed to handle and recover from hardware or software failures with minimal impact on users.

Methodology & Technology

Methodology

"The primary purpose of the software development industry is to deliver high-quality software within a limited budget, which requires proper planning for any software project. Various methodologies can be employed for this purpose, including:

- Agile
- Spiral
- Waterfall, etc.

For the development of the 'GigConnect' web application, we have decided to use the Agile methodology. Below is a more detailed explanation of the Agile methodology."

Agile Methodology

What is Agile Development Methodology?

Agile is one of the most commonly used methodologies in the software development field. It guides the development process by breaking it down into iterations or manageable increments known as sprints. Each sprint focuses on a specific feature, including its development, testing, and deployment. Agile methodology involves close collaboration between the development team and the client, ensuring that the client's needs and expectations are met.

Agile Model

Why is Agile Development Methodology Important?

Agile methodology is important because it is widely used in software development to deliver high-quality projects within the established timeline and budget. Below are some reasons why Agile methodology is important:

1. **Flexibility and Adaptability**
   Agile is designed to be flexible and adaptable. It embraces change throughout the development process, allowing team members to respond quickly to changes in requirements or priorities according to the client's needs.
2. **Improvement in Team Performance**
   Agile methodology promotes self-organizing teams, allowing members to take ownership of their work. This structure, combined with collaboration, often leads to higher team morale and job satisfaction.
3. **Reduced Risk and Increased Efficiency**
   A key success factor in Agile methodology is breaking down the project into smaller, manageable components among team members. This approach makes it easier to manage and deliver value at each step. Agile reduces the risk of large-scale project failure, as issues can be identified and addressed early in the development process, minimizing the impact of potential problems.

Pros and Cons of Agile Development Methodology

Pros of Agile methodology include:

- Timely Delivery
- Adaptability
- Ease of Collaboration
- High Profitability
- Less Preparatory Work

Cons of Agile methodology include:

- Transfer Difficulties
- Variable Goals
- Lack of Documentation
- Less Predictability

Why We Are Using Agile Development Methodology in "GigConnect" Web Application Development?

The Agile methodology is well-suited for the "GigConnect" project because it emphasizes flexibility, continuous feedback, and iterative progress. Here's why Agile is a good fit:

1. **Flexibility and Adaptability:** The project requirements may evolve as user feedback is gathered. Agile allows for changes to be seamlessly incorporated during the development process.
2. **Customer-Centric Development:** Agile's iterative cycles, or sprints, ensure that the product is developed with continuous input from stakeholders, aligning with the customer-focused nature of "GigConnect."
3. **Frequent Deliverables:** Agile allows the team to deliver functional parts of the app at the end of each sprint, enabling early testing and feedback from users.
4. **Collaboration and Communication:** The Agile approach promotes close collaboration between developers, designers, and project stakeholders, ensuring that the project remains aligned with the goal of connecting customers with gig workers.

Technology

PHP (Hypertext Preprocessor) PHP is a widely used, open-source scripting language suited for web development. It is especially useful for creating dynamic content on websites. PHP is embedded within HTML, making it easy to add functionality to web pages without needing to call external files. In the "GigConnect" project, PHP will handle server-side logic, process user requests, manage sessions, and interact with the database to provide seamless user experiences.

HTML (Hypertext Markup Language) HTML is the standard markup language for creating web pages and web applications. It forms the structure of the web pages in the "GigConnect" project, defining elements such as headings, paragraphs, links, and forms. HTML is fundamental for ensuring that content is displayed correctly on different devices and browsers.

CSS (Cascading Style Sheets) CSS is a stylesheet language used for describing the presentation of a document written in HTML. It controls the layout of multiple web pages all at once, ensuring a consistent look and feel across the "GigConnect" platform. In this project, CSS is crucial for designing responsive and visually appealing interfaces that enhance user experience.

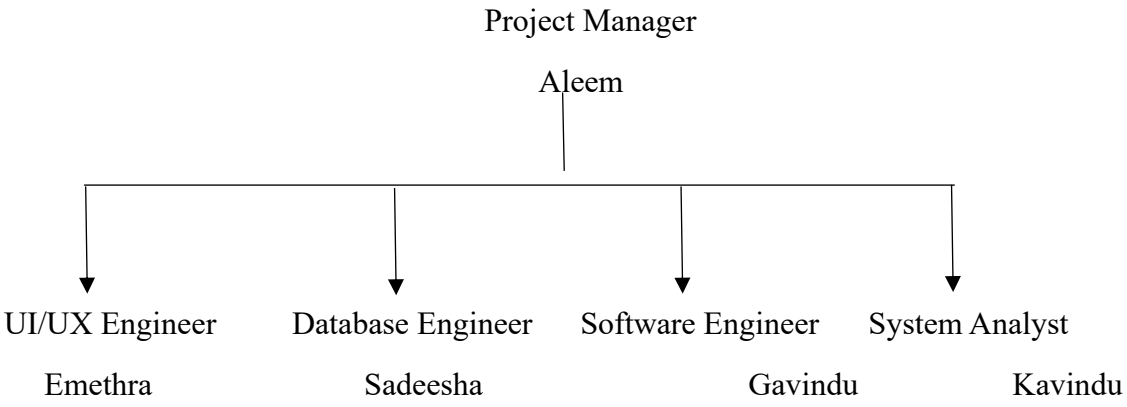
Bootstrap Bootstrap is a popular front-end framework for building responsive, mobile-first websites. It includes HTML, CSS, and JavaScript-based design templates for typography, forms, buttons, navigation, and other interface components. In "GigConnect," Bootstrap will be used to streamline the development process and ensure the website is responsive and accessible across various devices.

MySQL MySQL is a widely used open-source relational database management system (RDBMS). It will serve as the database backend for "GigConnect," storing all user data, job postings, bids, and reviews. MySQL's robustness and scalability make it an ideal choice for handling the project's data needs while ensuring data integrity and security.

Using Hardware and Software for this Project

- **Laptop/PC**: Development will primarily be conducted on laptops or desktop computers with adequate processing power (Intel i5/i7 or AMD Ryzen 5/7 with 8 GB RAM minimum).
- **Wi-Fi Router (Internet)**: A reliable internet connection is necessary for collaboration, testing, and deployment of the web application.
- **Secondary Storage Devices**: Portable hard disks and pen drives will be used for backup and transfer of project files.

Team Structure



Roles and Responsibility of the Team

"Roles & Responsibilities" refer to the specific duties, tasks, and areas of accountability that individuals or teams are expected to fulfil within a project or organization. These roles and responsibilities are typically defined to ensure clarity, efficiency, and effective collaboration throughout the development process, and ultimately the success of the software project.

Project Manager:

- Planning up the schedule and resource allocation for the project.
- Organizing team members.
- Cost estimation and creating an budget estimation chart.
- Managing reports and preparing necessary documents.
- Inspecting that the project is out of risk and the client needs and business goals are achieved.

UI/UX Engineer:

- User-Interface Designer.
- Creating Wireframes Diagram
- Ensuring Visual Consistency
- Designing Interactive Elements (Buttons, Navigation Menu, etc)

Software Engineer:

- Write and maintain clean, efficient, and scalable code for the application.
- Implement business logic and integrate third-party services if needed (without APIs, based on the project constraints).
- Debug and resolve technical issues encountered during development.
- Collaborate with other developers to ensure cohesive software architecture and functionality.

Database Engineer:

- Design and implement the database structure to support the application's data requirements.
- Optimize database queries and operations to ensure efficient data retrieval and storage.
- Manage database backups, migrations, and security protocols to safeguard user data.
- Work closely with the Backend Developer to integrate the database with server-side logic.

System Analyst:

- Analyze and understand the project requirements, translating them into technical pacifications.
- Assess system integration requirements and ensure compatibility with existing tools and platforms.
- Identify potential system improvements and propose solutions to optimize performance.
- Ensure that the system architecture aligns with the project's goals and objectives.

Cost

Project Name –“ GigConnect”

Date – 2024/08/30

WAGES

- **Project Manager** – 180,000 LKR
- **Frontend Developer** – 140,000 LKR
- **Backend Developer** – 160,000 LKR
- **UI/UX Engineer** – 130,000 LKR
- **Quality Assurance Engineer** – 120,000 LKR
  SUB TOTAL – 730,000 LKR

ESSENTIAL BILLS / OTHERS

- **Computer Maintenance** – 120,000 LKR
- **Internet Bill** – 45,000 LKR
- **Transportation** – 35,000 LKR
- **Electricity Bill** – 25,000 LKR
- **Other** – 15,000 LKR
  SUB TOTAL – 240,000 LKR

TOTAL COST

- **WAGES** – 730,000 LKR
- **ESSENTIAL BILLS** – 240,000 LKR
  TOTAL – 970,000 LKR

Hard Ware / Software Requirements

 Web Browsers:

- Google Chrome
- Microsoft Edge
- Mozilla Firefox

 Office Suite:

- Microsoft Office Package:
  - **MS Word** (for documentation)
  - **MS Excel** (for project planning)
  - **MS PowerPoint** (for presentations)
  - **MS Project** (for Gantt charts and project management)

 Operating System:

- **Windows 10** or **Windows 11** (latest version preferred for security and compatibility)

 Development Tools:

- **Code Editor**: Visual Studio Code (for writing and managing PHP, HTML, CSS, and Bootstrap code)
- **Local Server** WAMP (for running a local development environment with PHP and MySQL)
- **Version Control**: Git (for source code management)

 Other Tools:

- **Draw.io**: For creating diagrams and flowcharts related to project architecture and design.

Gantt Chart

Project plan

1. Initiation Phase (8 days)

Start Date: Thu 8/29/24
End Date: Mon 9/9/24

- **Objective:** Clearly define the project scope, goals, and deliverables. Identify the key stakeholders and gather initial requirements.
- **Tasks:**
    - **Define project objectives and deliverables:** Draft a document outlining project goals and expected outcomes.
    - **Identify key team roles and stakeholders:** Determine the project team and key stakeholders.
    - **Gather initial requirements:** Conduct meetings with stakeholders to collect initial business requirements.
    - **Budget planning:** Develop a preliminary project budget.
    - **Milestone Proposal:** Document the project proposal for stakeholder approval.
- **Milestone:** Project proposal approved.

2. Requirement Gathering (11 days)

Start Date: Thu 9/5/24
End Date: Mon 9/17/24

- **Objective:** Collect detailed requirements from stakeholders, focusing on both functional and non-functional needs.
- **Tasks:**
    - **Conduct interviews/surveys:** Engage with gig workers, contractors, and customers to gather input.

  - **Identify functional and non-functional requirements:** Document the system's required features and performance metrics.
  - **Create project charter:** Consolidate requirements and create a detailed project charter.
- **Milestone:** Project charter approved by stakeholders on **Wed 9/18/24**.

3. System Design Phase (7 days)

Start Date: Thu 9/19/24
End Date: Fri 9/27/24

- **Objective:** Translate requirements into technical and design solutions.
- **Tasks:**
  - **Create system architecture:** Design the system's overall architecture, considering scalability and security.
  - **Design user interface and experience:** Develop initial wireframes and mockups for the UI/UX.
  - **Prepare documentation:** Provide detailed technical and design documentation for the development team.
- **Milestone:** Design approved and locked in on **Fri 9/27/24**.

4. Development Phase (12 days)

Start Date: Fri 9/27/24
End Date: Fri 10/4/24

- **Objective:** Develop the database, back-end, and front-end components of the system.
- **Tasks:**
  - **Setup development environment:** Install necessary software, configure servers, and set up the coding environment.
  - **Database design:** Create and test the database schema.
  - **Front-end development:** Build the user interface based on wireframes.
  - **Back-end development:** Develop the server-side logic and APIs.
- **Milestone:** Complete core system development.

5. Testing and Quality Assurance (12 days)

Start Date: Wed 10/9/24
End Date: Tue 10/22/24

- **Objective:** Identify and resolve bugs; ensure that the system functions as intended.
- **Tasks:**
    - **Develop test cases:** Create unit, integration, and acceptance tests.
    - **Conduct testing:** Run test cases on the front and back end, covering all critical functionalities.
    - **Fix bugs:** Track, resolve, and retest bugs discovered during testing.
    - **User acceptance testing:** Perform testing with gig workers and customers.
- **Milestone:** System passes quality checks and is deemed ready for deployment on **Tue 10/22/24**.

6. Pre-Launch Deployment (7 days)

Start Date: Mon 10/23/24
End Date: Tue 10/31/24

- **Objective:** Finalize system setup and prepare for launch.
- **Tasks:**
    - **Add more content to the website:** Populate the website with necessary content and visuals.
    - **Check system accuracy:** Verify that all functionalities are operating as designed.
    - **Configure security and authentication:** Finalize security measures and user authentication processes.
    - **Final system testing:** Conduct end-to-end system tests.
    - **Training:** Provide system training for admin and workers.
- **Milestone:** System is fully configured and ready for public use on **Tue 10/31/24**.

7. Deployment Phase (4 days)

Start Date: Tue 10/31/24
End Date: Fri 11/4/24

- **Objective:** Deploy the system to the public.
- **Tasks:**
    - **Deploy website:** Make the website live for all users.

  - **Public notification:** Announce the launch to users (customers, workers, etc.).
  - **Monitor performance:** Closely observe the system performance for the first few days to ensure smooth operations.
- **Milestone:** Successful deployment of the platform.

8. Post-Launch Support and Maintenance (5 days)

Start Date: Fri 11/7/24
End Date: Fri 11/15/24

- **Objective:** Provide ongoing technical support and maintain system performance.
- **Tasks:**
  - **Monitor the system:** Track performance, bugs, and user feedback during the post-launch period.
  - **Provide technical support:** Offer support for user issues and system bugs.
  - **Issue resolutions:** Patch any bugs or issues that arise during user interactions.
- **Milestone:** Ensure system stability and support user base.

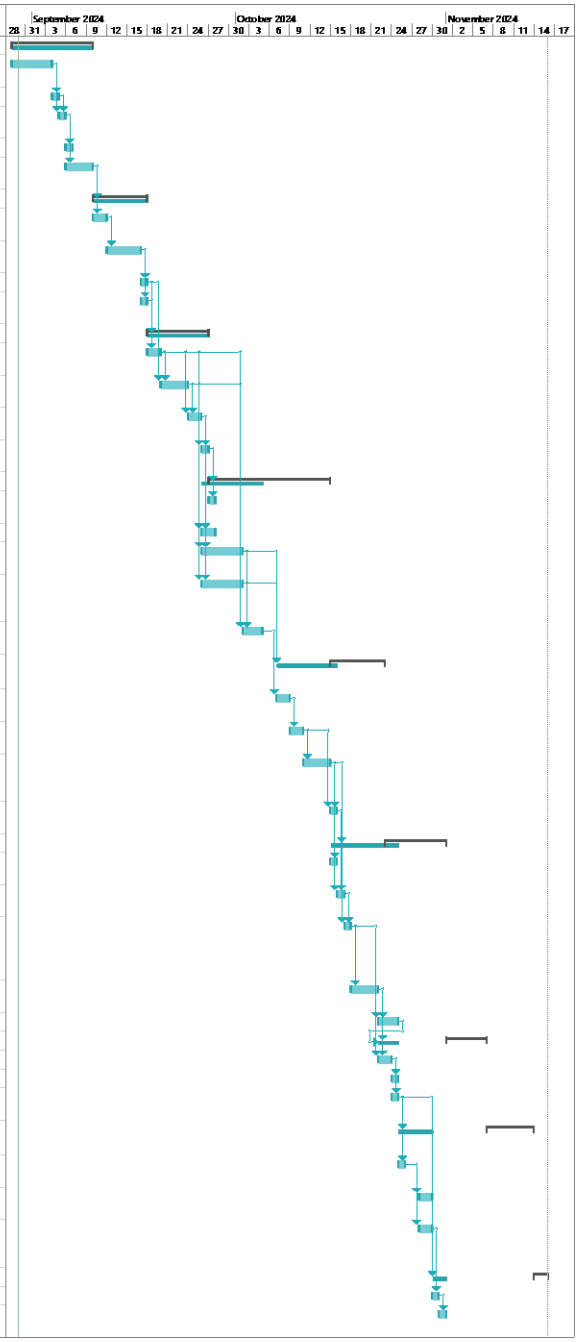
9. Project Closure (3 days)

Start Date: Mon 11/14/24
End Date: Fri 11/15/24

- **Objective:** Officially close the project and perform final reviews.
- **Tasks:**
  - **Finalize documentation:** Collect all deliverables, documents, and reports.
  - **Conduct a project review:** Hold a final meeting with stakeholders to review the project outcomes.
  - **Close contracts:** Officially close out the project and release team members.
- **Milestone:** Successful project closure on **Fri 11/15/24**.

Gantt Chart

ID	Task Mode	Task Name	Duration	Start	Finish	Predecessors
1		**01: Initiation Phase**	**8 days**	**Thu 8/29/24**	**Mon 9/9/24**	
2		Define project objective , scope and deliverables	4 days	Thu 8/29/24	Tue 9/3/24	
3		Identifying key teams and roles	1 day	Wed 9/4/24	Wed 9/4/24	2
4		Gathering initial requirements from stake holders	1 day	Thu 9/5/24	Thu 9/5/24	2,3
5		Budget planning	1 day	Fri 9/6/24	Fri 9/6/24	4
6		Milestone: project proposal Acceptance	2 days	Fri 9/6/24	Mon 9/9/24	4
7		**02: Requirement Gathering**	**6 days**	**Tue 9/10/24**	**Tue 9/17/24**	**6**
8		Conduct Interviews and surveys with gig workers	2 days	Tue 9/10/24	Wed 9/11/24	6
9		identify functional and non-functional requirements	3 days	Thu 9/12/24	Mon 9/16/24	8
10		Making project schedule	1 day	Tue 9/17/24	Tue 9/17/24	9
11		Milestone: project charter approval	1 day	Tue 9/17/24	Tue 9/17/24	9
12		**03: System Design**	**7 days**	**Wed 9/18/24**	**Thu 9/26/24**	**11**
13		Create system architecture and data models	2 days	Wed 9/18/24	Thu 9/19/24	10,11
14		Design the user interface and user experience	2 days	Fri 9/20/24	Mon 9/23/24	10,13
15		Create wireframes and mockup diagrams	2 days	Tue 9/24/24	Wed 9/25/24	13,14
16		Milestones: Acceptance of produced system architecture	1 day	Thu 9/26/24	Thu 9/26/24	14,15
17		**04: Development Phase**	**12 days**	**Fri 9/27/24**	**Mon 10/14/24**	**16**
18		Setup development environment	1 day	Fri 9/27/24	Fri 9/27/24	16
19		Creating Database	2 days	Thu 9/26/24	Fri 9/27/24	13,14,15
20		Create user friendly interface (front end development)	4 days	Thu 9/26/24	Tue 10/1/24	14,15
21		Database setup, Server-side scripting (Back end development)	4 days	Thu 9/26/24	Tue 10/1/24	13,14,15
22		Testing Front end and back end interactions	3 days	Wed 10/2/24	Fri 10/4/24	21,13,14,20
23		**05: Testing and quality assurance**	**6 days**	**Tue 10/15/24**	**Tue 10/22/24**	**21,20**
24		Develop test cases based on requirements	2 days	Mon 10/7/24	Tue 10/8/24	22
25		Conduct unit testing, intergration and system testing	2 days	Wed 10/9/24	Thu 10/10/24	24
26		Conduct user acceptance testing with gig workers members	2 days	Fri 10/11/24	Mon 10/14/24	25
27		Milestone: Identify and solve bugs or issues	1 day	Tue 10/15/24	Tue 10/15/24	25,26
28		**06: Pre-Launch deployment**	**7 days**	**Wed 10/23/24**	**Thu 10/31/24**	**27**
29		Adding more content into the webiste	1 day	Tue 10/15/24	Tue 10/15/24	26
30		Checking the accuracy of the inserted content	1 day	Wed 10/16/24	Wed 10/16/24	26,27
31		Setting up necessary infrastructure, system and processes to ensure smooth launch.	1 day	Thu 10/17/24	Thu 10/17/24	26,30
32		Server configurating, testing functionalities	2 days	Fri 10/18/24	Mon 10/21/24	31
33		Staff training	3 days	Tue 10/22/24	Thu 10/24/24	31,32
34		**07: Deployment Phase**	**4 days**	**Fri 11/1/24**	**Wed 11/6/24**	**32,33**
35		Test entire system	2 days	Tue 10/22/24	Wed 10/23/24	31,32
36		Develop user manuals	1 day	Thu 10/24/24	Thu 10/24/24	35
37		Milestone: Deploy the gig worker's website	1 day	Thu 10/24/24	Thu 10/24/24	35
38		**08: Post launch support and maintance**	**5 days**	**Thu 11/7/24**	**Wed 11/13/24**	**37**
39		Provide ongoing technical support	1 day	Fri 10/25/24	Fri 10/25/24	37
40		Monitor website and address any issues or bags if arisen	2 days	Mon 10/28/24	Tue 10/29/24	39
41		implement regular updates based on user feedback and market trends	2 days	Mon 10/28/24	Tue 10/29/24	39
42		**09: Project Closure**	**2 days**	**Thu 11/14/24**	**Fri 11/15/24**	**37**
43		Finalize all project documents.	1 day	Wed 10/30/24	Wed 10/30/24	41
44		Conduct a project review with stake holders	1 day	Thu 10/31/24	Thu 10/31/24	43



The project is planned from **August 29, 2024, to November 15, 2024**, spanning **approximately 2.5 months**. The Gantt chart is divided into **nine major phases**, each containing tasks that are associated with specific start and end dates. These tasks also have dependencies shown by the arrows connecting them, indicating that certain tasks cannot begin until others are completed.

Here are the most critical risks in the project,

1. **Scope Creep:** Poorly defined objectives may lead to expanding the project beyond its initial plan, causing delays and increased costs.

2. **Incomplete Requirements:** Misunderstood or missing requirements could result in developing incorrect features, leading to costly rework.
3. **Integration Issues:** Problems integrating front-end, back-end, and databases could delay development and create bugs.
4. **Insufficient Testing:** Lack of thorough testing may allow critical bugs to slip through, resulting in system failures after launch.
5. **Deployment Downtime:** Technical issues during deployment could cause system downtime, harming user experience.
6. **Post-Launch Bugs:** Undetected issues in the final stages could disrupt the user experience and require urgent fixes.

Addressing these risks early on can prevent major disruptions in the project timeline and quality.