System Architecture Diagram

Here is an end-to-end map of the data flow through the system for the TASKIFY platform. This diagram includes the front-end, back-end, database, and external services.

```
graph TD
  A[Client (Browser)]
  B[Web Server (Node.js + Express)]
  C[Front-end (React)]
  D[Back-end (Node.js + Express)]
  E[Database (MySQL)]
  F[Authentication Service (Auth0)]
  G[Payment Gateway (Stripe)]
  H[File Storage (AWS S3)]
  I[Notification Service (Twilio)]

  A --> C
  C --> B
  B --> D
  D --> E
  D --> F
  D --> G
  D --> H
  D --> I
```

Client (Browser): The user interface where clients and freelancers interact with the platform.

Web Server (Node.js + Express): Handles client requests and serves the React application.

Front-end (React): The user-facing part of the application, providing an interactive UI.

Back-end (Node.js + Express): Manages business logic, processes requests, and interacts with the database and external services.

Database (MySQL): Stores user data, project details, transaction history, etc.

Authentication Service (Auth0): Manages user authentication and authorization.

Payment Gateway (Stripe): Handles payment processing between clients and freelancers.

File Storage (AWS S3): Stores files such as user profile pictures, project attachments, etc.

Notification Service (Twilio): Sends notifications to users via SMS or email.

APIs and Methods

Web Client API Routes

/api/users

GET: Returns the user's information based on session id
POST: Registers a new user with provided details
PUT: Updates user information
DELETE: Deletes a user account

/api/projects

GET: Returns a list of projects based on filter criteria
POST: Creates a new project with provided details

PUT: Updates project information
DELETE: Deletes a project
/api/bids

GET: Returns a list of bids for a specific project
POST: Submits a new bid for a project
PUT: Updates bid information
DELETE: Deletes a bid
/api/payments

POST: Initiates a payment transaction between client and freelancer
GET: Retrieves transaction history for a user
/api/auth

POST: Authenticates a user and returns a token
GET: Verifies the authentication token
External Client API Endpoints
/api/notifications

POST: Sends a notification to a user via Twilio
Parameters:
userId (string): The ID of the user to notify
message (string): The notification message content
/api/files

POST: Uploads a file to AWS S3
Parameters:
file (file): The file to be uploaded
userId (string): The ID of the user uploading the file
GET: Retrieves a file URL from AWS S3
Parameters:
fileId (string): The ID of the file to retrieve
/api/integrations

POST: Integrates with a third-party service
Parameters:
service (string): The name of the third-party service
data (object): Data required for integration
3rd Party APIs
Auth0

POST /oauth/token: To obtain an access token for authenticating users
GET /userinfo: To retrieve user profile information
Stripe

POST /v1/charges: To create a new charge

GET /v1/charges/{charge_id}: To retrieve a specific charge

POST /v1/refunds: To issue a refund

Twilio

POST /2010-04-01/Accounts/{AccountSid}/Messages.json: To send an SMS message

GET /2010-04-01/Accounts/{AccountSid}/Messages.json: To retrieve sent messages

Data Model

A data model diagram to clarify how data will be stored:

```
erDiagram
  USER {
    int id
    string username
    string email
    string password_hash
    string profile_picture_url
    datetime created_at
    datetime updated_at
  }

  PROJECT {
    int id
    int user_id
    string title
    string description
    float budget
    string status
    datetime created_at
    datetime updated_at
  }

  BID {
    int id
    int project_id
    int user_id
    float amount
    string message
    datetime created_at
    datetime updated_at
  }

  PAYMENT {
    int id
    int project_id
    int user_id
    float amount
    string status
    datetime created_at
    datetime updated_at
  }

  USER ||--o{ PROJECT : "creates"
  PROJECT ||--o{ BID : "has"
  PROJECT ||--o{ PAYMENT : "includes"
  USER ||--o{ BID : "submits"
  USER ||--o{ PAYMENT : "receives"
```

User Stories

User Registration and Login

As a new user, I want to register an account with my email and password, so that I can access the platform.
As a registered user, I want to log in with my email and password, so that I can access my account and use the platform.

Project Creation and Management

As a client, I want to create a new project by providing details such as title, description, and budget, so that I can find freelancers to complete the project.
As a client, I want to update or delete my projects, so that I can manage my postings effectively.

Bidding on Projects

As a freelancer, I want to browse available projects and submit bids, so that I can offer my services to potential clients.
As a freelancer, I want to update or withdraw my bids, so that I can manage my applications.

Payment Processing

As a client, I want to initiate payments to freelancers upon project completion, so that I can compensate them for their work.
As a freelancer, I want to receive payments for completed projects, so that I can get paid for my work.

Notifications and Communication

As a user, I want to receive notifications about important events (e.g., new bids, project updates), so that I can stay informed about my activities on the platform.
As a user, I want to send and receive messages with other users, so that I can communicate effectively about projects.

Mockups

Include mockups of each view that will need to be created for your MVP. Below are the primary views:

Home Page

Welcome message
Login/Register buttons
Featured projects
User Dashboard

User profile information
List of active projects
Notifications panel
Project Creation

Form to enter project details (title, description, budget)
Submit button to create a new project
Project Listing

List of available projects with filters for category, budget, and status
Button to view project details and submit a bid
Bid Submission

Form to enter bid amount and message
Submit button to place a bid on a project
Payment Processing

Form to enter payment details
Submit button to process payment
Using a tool like Balsamiq, we shall create wireframes for these views to visualize the user interface and ensure a good user experience.

WEEK 2 PROGRESS

Weekly Progress Rating

Progress Rating: 7/10

Explanation of Progress Assessment

This week, substantial progress has been made on several fronts of the TASKIFY platform development. The team successfully completed major milestones, including setting up the development environment, implementing core functionalities, and creating initial API endpoints. However, there were a few areas where progress was slower than anticipated, primarily due to unexpected technical challenges and resource constraints.

Completed Parts of the Project

1. **Development Environment Setup**
   - The development environment has been configured, including version control (Git), continuous integration (CI) pipelines, and deployment strategies.
   - Branching and merging strategies have been established to ensure smooth collaboration among team members.
2. **Core Functionality Implementation**
   - User registration and authentication using Auth0 have been fully implemented.
   - Project creation and management features have been developed, allowing clients to post and manage their projects.
   - Initial API routes for user and project management have been created and tested.
3. **Database Schema Design**
   - The MySQL database schema has been designed and implemented, including tables for users, projects, bids, and payments.
   - Data model relationships have been established, ensuring data integrity and consistency.
4. **Front-End Development**
   - The initial user interface has been designed using React, including key pages such as the home page, user dashboard, and project listing.
   - Basic navigation and layout have been implemented, providing a foundation for further development.

Incomplete Aspects of the Project

1. **Bidding System**
   - The functionality for freelancers to submit bids on projects is partially complete. The backend logic is implemented, but the front-end integration and UI design are still in progress.
   - Bid management features, including updating and withdrawing bids, need to be finalized.

2. **Payment Processing**
   - Integration with the Stripe API for handling payments is underway but not yet complete. The backend endpoints are being developed, and the front-end forms need to be created.
   - Security measures for payment transactions are being reviewed and implemented.
3. **Notifications and Communication**
   - The notification system using Twilio is still in the development phase. Initial setup and configuration are done, but integration with the platform and user interface design are pending.
   - Messaging functionality between clients and freelancers is planned but has not been started.
4. **Testing and Quality Assurance**
   - Automated testing scripts are partially written, but comprehensive testing across all modules is needed to ensure robustness and reliability.
   - User acceptance testing (UAT) is scheduled for the coming weeks to gather feedback and identify potential issues.

Next Steps

To continue progressing towards the project goals, the team will focus on the following tasks in the coming week:

1. Complete the bidding system and integrate it with the front-end.
2. Finalize payment processing with Stripe and ensure secure transactions.
3. Develop and integrate the notification system with Twilio.
4. Enhance testing coverage and conduct UAT to identify and fix any issues.
5. Continue refining the user interface and improving the user experience.

CHALLENGES

Challenges

Technical Challenge

The most difficult technical challenge encountered this week was the integration of the Stripe API for payment processing.

The complexity of integrating a secure and reliable payment gateway posed several issues:

1. **Understanding Stripe's API Documentation:**
   - Stripe's API documentation is extensive and covers a wide range of features. Parsing through the documentation to identify the necessary endpoints and

integration steps was time-consuming. Each aspect of the payment process, from tokenization to handling webhooks for payment confirmation, required careful consideration and implementation.

2. **Security Concerns:**
   - Ensuring the security of financial transactions is paramount. We had to implement robust security measures to protect sensitive data, such as payment information and user details. This involved setting up secure servers with HTTPS, using environment variables to store API keys securely, and implementing proper error handling and logging mechanisms.
3. **Handling Different Payment Scenarios:**
   - Our platform needs to support various payment scenarios, including one-time payments, recurring payments, and refunds. Each scenario had unique requirements and edge cases that needed to be addressed. This included creating customer objects, managing subscriptions, and handling disputes or chargebacks.
4. **Testing the Integration:**
   - Thorough testing was essential to ensure the integration worked seamlessly. We set up a staging environment to test the entire payment flow, from capturing payment details to processing transactions and handling webhooks. Simulating different scenarios and ensuring consistent behavior across all tests proved challenging.
5. **Ensuring User Experience:**
   - The payment process needed to be user-friendly and seamless. Designing intuitive front-end forms and ensuring they communicated correctly with the backend was crucial. We had to balance functionality with simplicity to ensure users had a smooth experience.

To overcome these challenges, we divided the work among team members, with each person focusing on a specific aspect of the integration. Regular team meetings and code reviews helped ensure alignment and address issues promptly. Utilizing Stripe's support resources and community forums also provided valuable insights and solutions to some of the problems we encountered.

Non-Technical Challenge

The most difficult non-technical challenge encountered this week was coordinating effective communication and collaboration within the team.

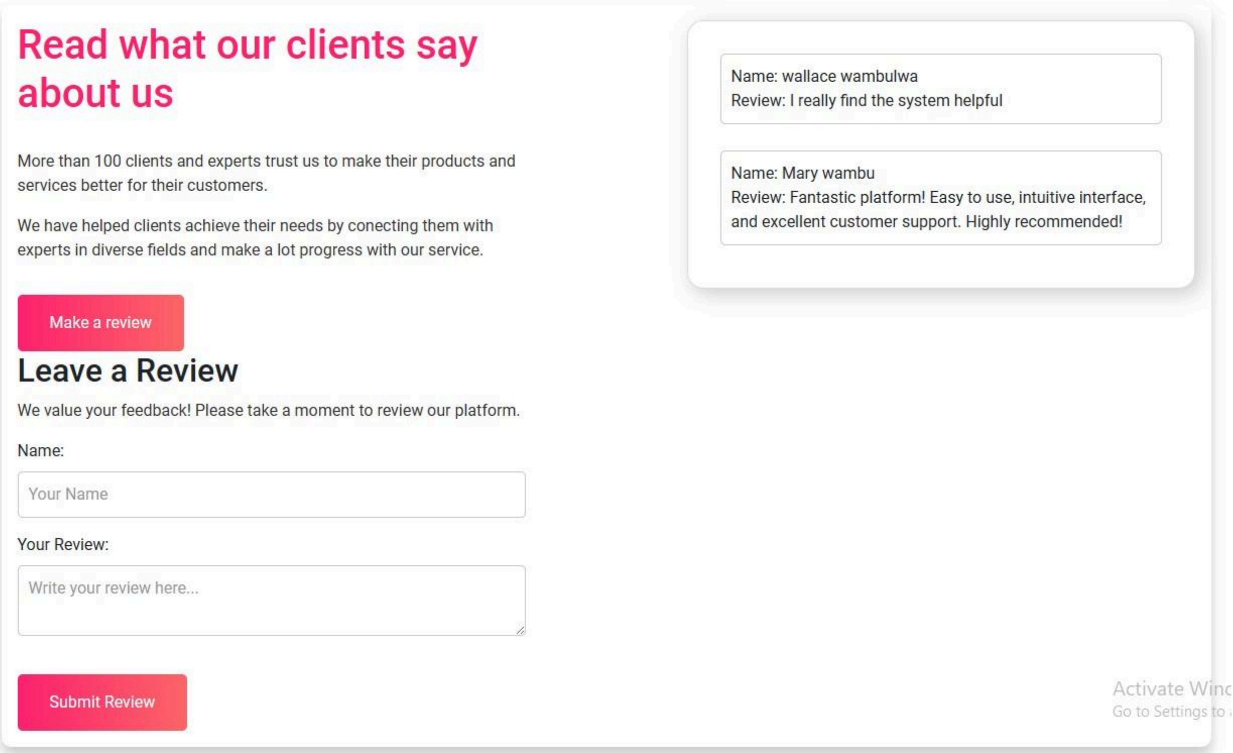
1. **Remote Collaboration:**
   - Our team is distributed across different time zones, which made synchronous communication difficult. Scheduling meetings that accommodated everyone's availability required careful planning and often led to delays in decision-making and progress updates.
2. **Aligning Goals and Priorities:**

- Ensuring that all team members were aligned on project goals and priorities was challenging. Different perspectives and interpretations of the project requirements sometimes led to misunderstandings and conflicting approaches. We needed to establish clear and consistent communication channels to align everyone's efforts.

3. **Managing Workloads and Responsibilities:**
    - Balancing workloads and distributing responsibilities fairly among team members was crucial to maintain productivity and morale. Some team members felt overwhelmed with their tasks, while others needed more direction. Regular check-ins and progress updates helped identify and redistribute tasks to ensure a more even workload.
4. **Maintaining Motivation and Engagement:**
    - Keeping the team motivated and engaged, especially during challenging phases of the project, was essential. Working remotely can sometimes lead to feelings of isolation and detachment from the team. We implemented strategies such as virtual team-building activities and regular feedback sessions to foster a sense of community and keep morale high.
5. **Documenting and Sharing Knowledge:**
    - Ensuring that all team members had access to the necessary information and resources was vital for smooth collaboration. We encountered challenges in documenting and sharing knowledge effectively, especially with the fast-paced nature of the project. Utilizing collaborative tools such as Confluence for documentation and Slack for real-time communication helped address this issue.

To address these challenges, we established a structured communication plan that included regular stand-up meetings, progress updates, and dedicated channels for specific discussions. We also emphasized the importance of documentation and created shared repositories for resources and project-related information. These measures helped improve coordination, ensure alignment, and maintain a productive and cohesive team environment.

SCREENSHOTS



Read what our clients say about us

More than 100 clients and experts trust us to make their products and services better for their customers.

We have helped clients achieve their needs by conecting them with experts in diverse fields and make a lot progress with our service.

Leave a Review

We value your feedback! Please take a moment to review our platform.

Name:

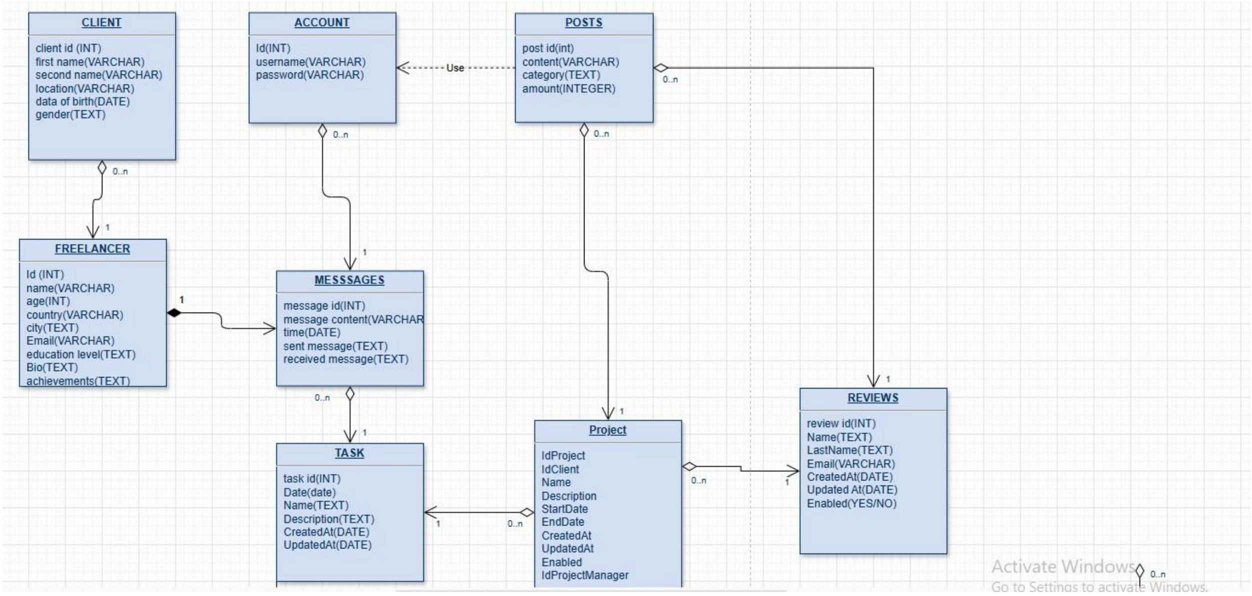
Your Name

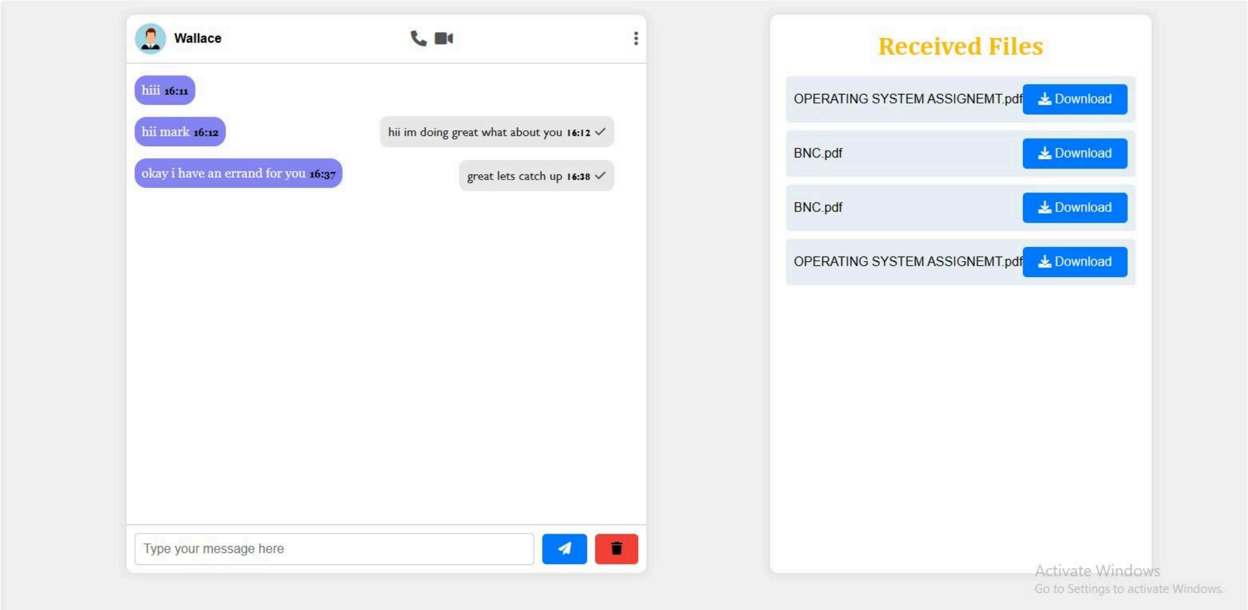
Your Review:

Write your review here...

Name: wallace wambulwa
Review: I really find the system helpful

Name: Mary wambu
Review: Fantastic platform! Easy to use, intuitive interface, and excellent customer support. Highly recommended!





Wallace

hiii 16:11

hii mark 16:12

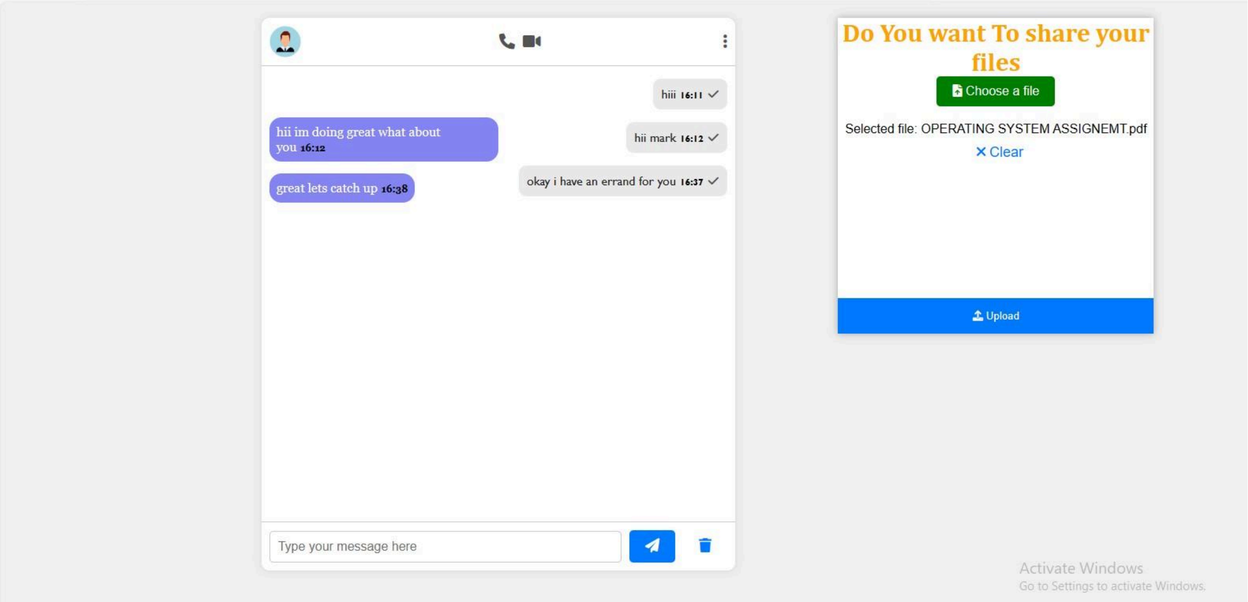
hii im doing great what about you 16:12 ✓

okay i have an errand for you 16:37

great lets catch up 16:38 ✓

Received Files

OPERATING SYSTEM ASSIGNEMT.pdf	Download
BNC.pdf	Download
BNC.pdf	Download
OPERATING SYSTEM ASSIGNEMT.pdf	Download



hiii 16:11 ✓

hii im doing great what about you 16:12

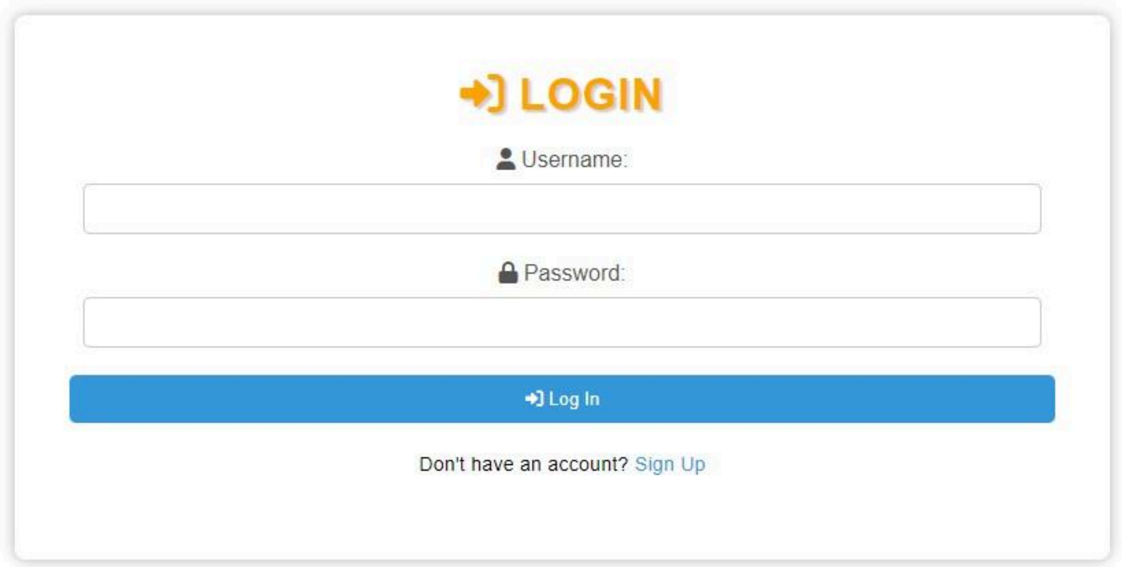
hii mark 16:12 ✓

okay i have an errand for you 16:37 ✓

great lets catch up 16:38

Do You want To share your files

Selected file: OPERATING SYSTEM ASSIGNEMT.pdf

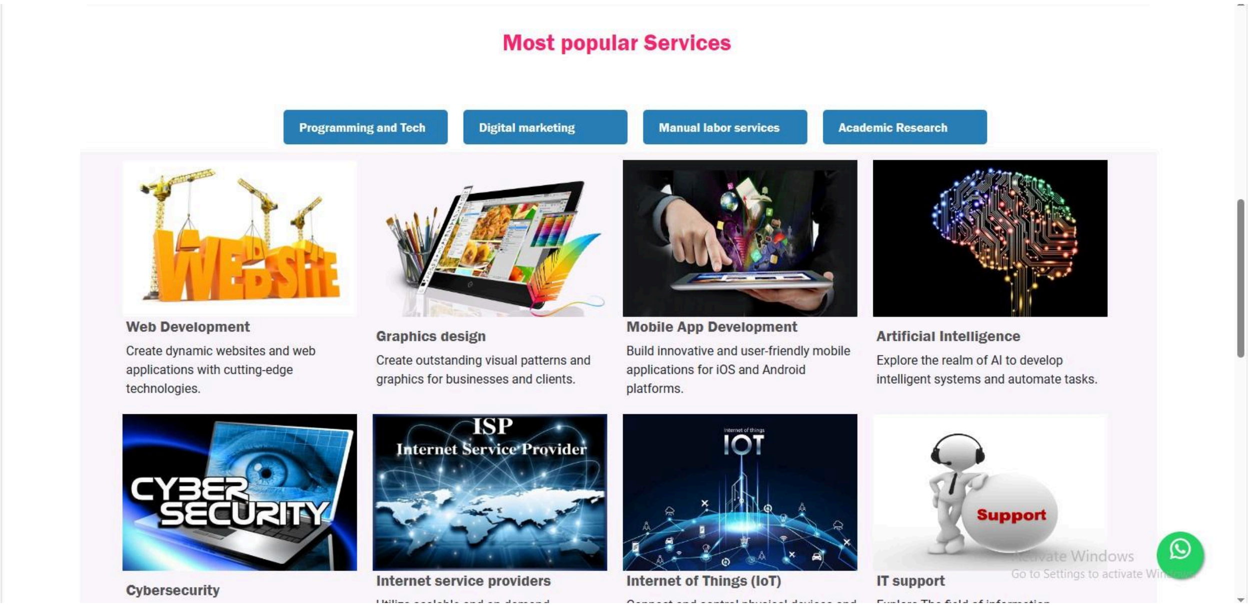


LOGIN

Username:

Password:

Don't have an account? Sign Up



Most popular Services

Web Development

Create dynamic websites and web applications with cutting-edge technologies.

Graphics design

Create outstanding visual patterns and graphics for businesses and clients.

Mobile App Development

Build innovative and user-friendly mobile applications for iOS and Android platforms.

Artificial Intelligence

Explore the realm of AI to develop intelligent systems and automate tasks.

Cybersecurity

Internet service providers

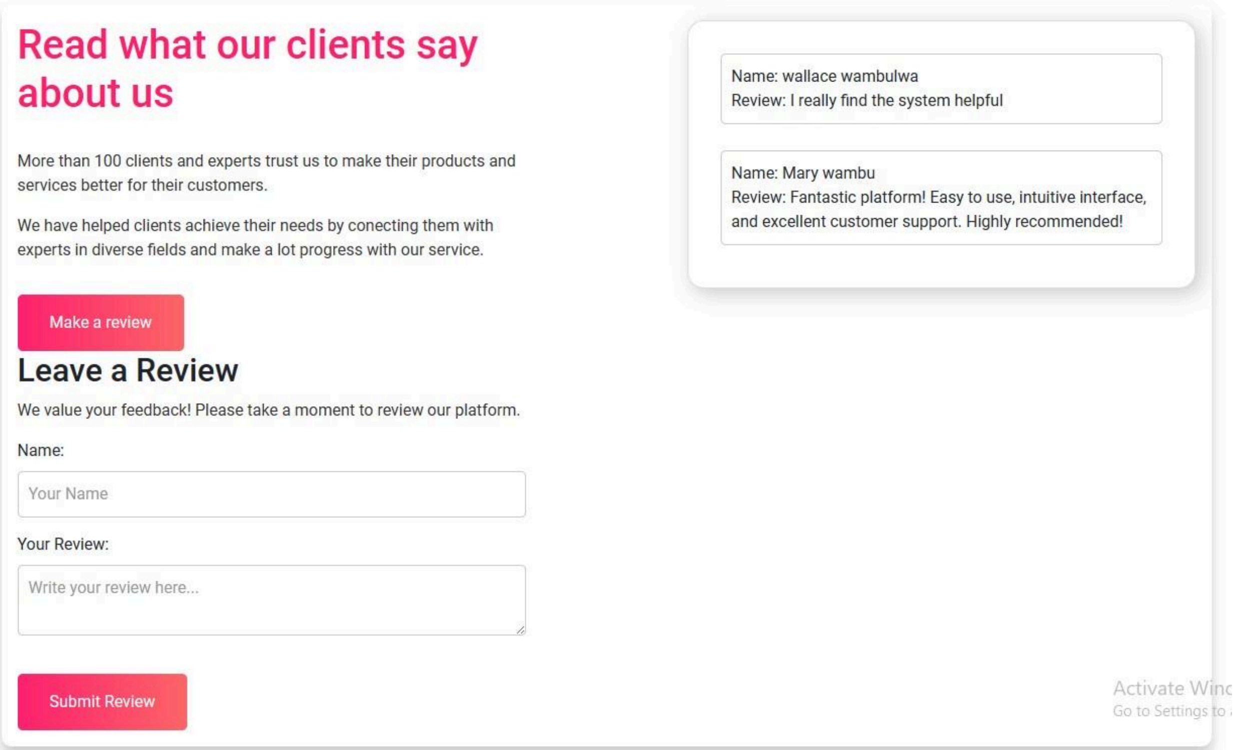
Utilize scalable and on-demand

Internet of Things (IoT)

Connect and control physical devices and

IT support

Explore The field of information



Read what our clients say about us

More than 100 clients and experts trust us to make their products and services better for their customers.

We have helped clients achieve their needs by conecting them with experts in diverse fields and make a lot progress with our service.

Make a review

Name: wallace wambulwa
Review: I really find the system helpful

Name: Mary wambu
Review: Fantastic platform! Easy to use, intuitive interface, and excellent customer support. Highly recommended!

Leave a Review

We value your feedback! Please take a moment to review our platform.

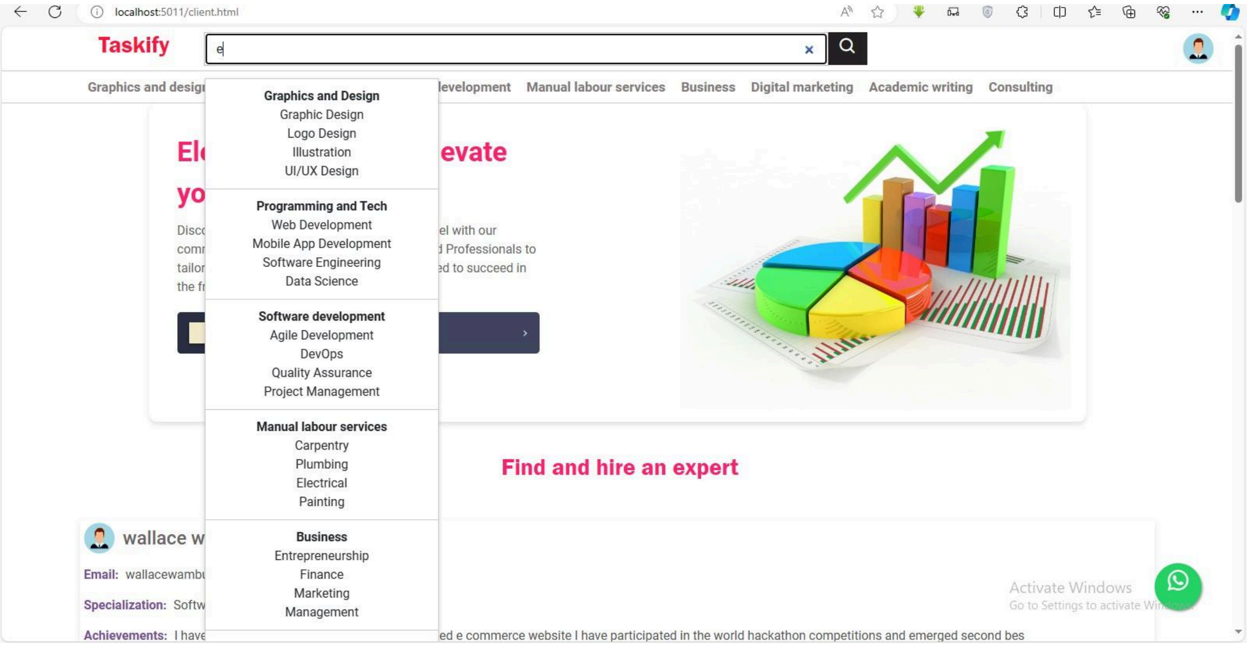
Name:

Your Name

Your Review:

Write your review here...

Submit Review



localhost:5011/client.html

Taskify

e ×

Graphics and desig levelopment Manual labour services Business Digital marketing Academic writing Consulting

Graphics and Design
Graphic Design
Logo Design
Illustration
UI/UX Design

Programming and Tech
Web Development
Mobile App Development
Software Engineering
Data Science

Software development
Agile Development
DevOps
Quality Assurance
Project Management

Manual labour services
Carpentry
Plumbing
Electrical
Painting

Business
Entrepreneurship
Finance
Marketing
Management

El evate
yo

Disc el with our
comr d Professionals to
tailor ed to succeed in
the fr

Find and hire an expert

wallace w

Email: wallacewambu

Specialization: Softw

Achievements: I have ed e commerce website I have participated in the world hackathon competitions and emerged second bes