A
PROJECT REPORT ON

“TaskBuddy”

SUBMITTED IN
PARTIAL FULFILLMENT OF

DIPLOMA IN ADVANCED COMPUTING (PG-DAC)

BY

Mayur Gadakh,
Prathamesh Hendre,
Abhay Siddheshware,
Abhishekh Pawar

UNDER THE GUIDENCE OF

Omkar Chougale

AT

SUNBEAM INSTITUTE OF INFORMATION TECHNOLOGY,
Pune

SUNBEAM INSTITUTE OF INFORMATION TECHNOLOGY, PUNE

CERTIFICATE

This is to certify that the project

TaskBuddy

Has been submitted by

Mayur Gadakh,
Prathamesh Hendre,
Abhay Siddheshware,
Abhishekh Pawar

In partial fulfillment of the requirement for the Course of **PG Diploma in Advanced Computing (PG-DAC AUG2015)** as prescribed by The **CDAC** ACTS, PUNE.

Place: Pune

Date: 30-JAN-2016

<Omkar Chougale>
Project Guide

<Alumni Mentor Name>
Alumni Mentor

ACKNOWLEDGEMENT

A project usually falls short of its expectation unless aided and guided by the right persons at the right time. We avail this opportunity to express our deep sense of gratitude towards Mr. Prashant Lad (Director, SIIT Karad), Pratik Ninganukar(Course Coordinator, SIIT, Karad) and Mr. Omkar Chougale (Our Project Guide ,SIIT Karad).

We are deeply indebted and grateful to them for their guidance, encouragement and deep concern for our project. Without their critical evaluation and suggestions at every stage of the project, this project could never have reached its present form.

Last but not the least we thank the entire faculty, especially Omkar Chougale and the staff members of Sunbeam Institute of Information Technology, Karad for their support.

TEAM SUNBEAM

DAC August 2024 Batch,

PG-DAC Karad.

ABSTRACT

TaskBuddy is an **on-demand service platform** designed to connect customers with skilled taskers for various household errands and professional services. The system provides a **seamless booking experience**, allowing users to hire taskers for services such as **on-demand driving, moving assistance, home repairs, and more**. Taskers, in turn, can register on the platform to offer their expertise, manage their availability, and earn money by completing tasks efficiently.

The project incorporates essential features such as **user authentication, task booking, real-time notifications, secure payment processing, and task completion tracking**. TaskBuddy ensures a **user-friendly experience** with a **well-structured database, optimized search functionality, and an intuitive interface**. Security measures, including **role-based access control, data encryption, and secure transactions**, are implemented to protect user data and maintain platform integrity.

The system is designed to be **scalable, reliable, and high-performing**, capable of handling a growing number of users and transactions. Through rigorous **testing and validation**, TaskBuddy has been optimized for usability, performance, and security. This project aims to **streamline the task outsourcing process**, making it more **efficient, convenient, and accessible** for both customers and taskers. Future enhancements may include **AI-based tasker recommendations, mobile app integration, and multilingual support** to further improve accessibility and service efficiency.

INDEX

LIST OF TABLES

LIST OF FIGURES

Introduction of Project

TaskBuddy is an innovative web service designed to connect customers with skilled taskers for a variety of household errands and professional services. Whether users need assistance with moving, on-demand driving, home repairs, or other specialized tasks, TaskBuddy offers a seamless platform to find and book reliable service providers. The platform ensures convenience by allowing customers to browse taskers based on skills, location, availability, and ratings, making it easier to find the right person for the job.

One of the standout features of TaskBuddy is its advanced booking system, which allows customers to schedule services in advance while ensuring real-time availability updates. Taskers can also register on the platform, set their rates, and manage bookings efficiently, providing them with an opportunity to earn money based on their expertise. A secure payment gateway, transparent pricing, and a review system further enhance the user experience, ensuring trust and reliability in every transaction.

TaskBuddy is built with a user-friendly interface that simplifies the process of booking and managing services. Customers can effortlessly search for taskers based on categories, ensuring they find the right professional for their specific needs. The platform also provides taskers with a personalized dashboard to track their earnings, manage schedules, and communicate with clients. By incorporating real-time notifications and reminders, TaskBuddy enhances efficiency for both customers and taskers, reducing delays and ensuring smooth service execution.

Security and reliability are at the core of TaskBuddy's operations. The platform includes identity verification for taskers, secure payment processing, and a transparent rating and review system to foster trust among users. With a focus on customer satisfaction, TaskBuddy continuously improves its features based on user feedback, making it a dependable solution for everyday tasks. Whether for professional services, household chores, or religious bookings, TaskBuddy streamlines the process, offering a convenient and efficient way to get things done.

PRODUCT OVERVIEW AND SUMMARY

2.1 Purpose

The purpose of the **TaskBuddy** project is to create a seamless platform that connects customers with skilled taskers for various household and professional services. Many individuals struggle to find reliable help for tasks such as on-demand driving, packing for a move, or general home maintenance. TaskBuddy bridges this gap by providing a trusted web service where customers can easily search for, book, and manage taskers based on their specific needs. The platform ensures convenience, efficiency, and security by allowing users to schedule services in advance and track task progress in real time.

Additionally, TaskBuddy empowers taskers by offering them a flexible way to earn money through their skills. By providing a structured environment for service booking, payment processing, and customer reviews, the platform ensures a transparent and professional experience for both customers and taskers. The inclusion of features such as identity verification, secure transactions, and a rating system fosters trust and reliability. Whether for everyday chores, skilled services, or religious bookings like Sai Darshan tickets, TaskBuddy simplifies the process, making it easier for users to get things done efficiently.

2.2 Scope

The scope of **TaskBuddy** extends to providing a comprehensive and user-friendly platform that facilitates seamless connections between customers and taskers for various household and skilled services. The system enables users to find, book, and schedule services such as on-demand driving, packing for a move, home maintenance, and more. Customers can browse through available taskers based on their skills, ratings, and availability, ensuring they get the best service possible. The platform also incorporates an integrated payment system for secure transactions, along with real-time notifications for booking updates and task progress tracking.

The platform aims to support both **individual customers and businesses**, making it scalable for future enhancements like additional service categories, AI-driven task recommendations, and multilingual support. By ensuring a **secure, efficient, and flexible** environment for both customers and taskers, TaskBuddy sets the foundation for a **trusted gig economy** platform, enabling convenience, employment opportunities, and an improved service experience.

2.3 User Classes and Characteristics

The **TaskBuddy** system is designed to cater to multiple user groups, each with distinct roles, functionalities, and interactions within the platform. The primary user classes include **Customers, Taskers, and Administrators**, each with specific characteristics and requirements.

1. **Customers**
    - Customers are individuals or businesses seeking taskers for various services such as home maintenance, on-demand driving, and moving assistance.
    - They can register on the platform, search for taskers, book services, make payments, and leave reviews.
    - Customers may vary in technical proficiency, requiring a user-friendly interface with guided navigation.
    - Some customers may book religious services like **Sai Darshan tickets**, making accessibility and seamless booking features essential.
2. **Taskers**
    - Taskers are skilled professionals or freelancers who offer services based on their expertise.
    - They can create profiles, set availability, accept or reject bookings, and receive payments for completed tasks.
    - Taskers may work part-time or full-time, requiring flexible scheduling options and efficient communication tools.
    - A rating and review system helps taskers build credibility and attract more bookings.
3. **Administrators**
    - Admins oversee the platform's operations, ensuring smooth transactions, resolving disputes, and managing users.
    - They have access to dashboards for monitoring bookings, payments, and system analytics.
    - Admins may also handle customer support requests and enforce security measures to prevent fraudulent activities

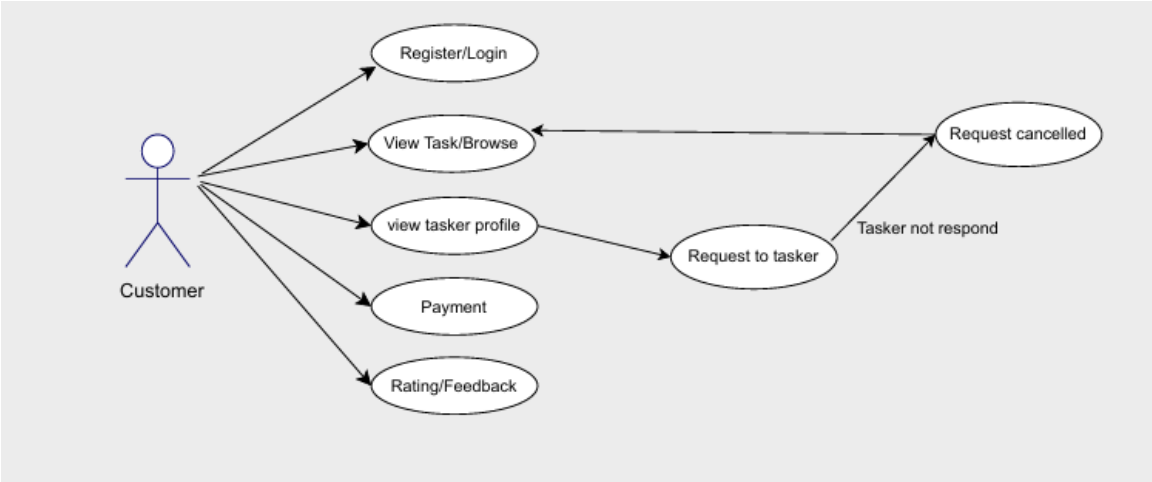
2.4 Design and Implementation Constraints

The **TaskBuddy** system must adhere to several design and implementation constraints to ensure **scalability, security, performance, and usability** while meeting project requirements. These constraints impact the development, deployment, and overall functionality of the platform.

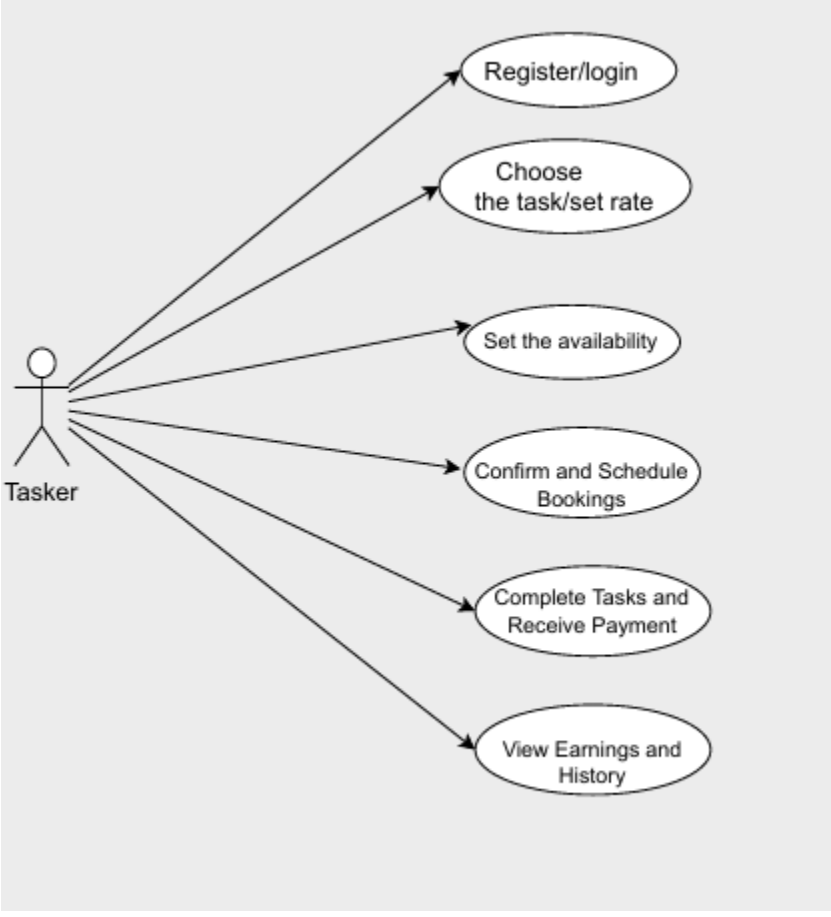
1. **Technology Stack Constraints**
    - The system will be developed using **web technologies** such as **React.js** for the MS.Net for the backend.
    - The database will be **MS SQL**, depending on scalability needs.
2. **Security Constraints**
    - User authentication should follow **OAuth 2.0** or **JWT-based token authentication** to prevent unauthorized access.
    - Role-based access control (RBAC) must be implemented to restrict actions based on user roles (Customer, Tasker, Admin).
3. **Performance Constraints**
    - The system should handle **high traffic loads** and concurrent user interactions without performance degradation.
    - API response time should be optimized to be within **200-500 milliseconds** for smooth user experience.
    - The database should support **indexed searches** to allow quick retrieval of taskers based on filters like location, skills, and availability.
4. **Usability Constraints**
    - The platform must be **mobile-responsive**, ensuring accessibility on desktops, tablets, and smartphones.
    - The UI should be designed with **ease of navigation**, catering to users with minimal technical knowledge.
    - Booking and payment processes should require **minimal steps** to reduce user drop-off rates.

Use-Case Diagram:

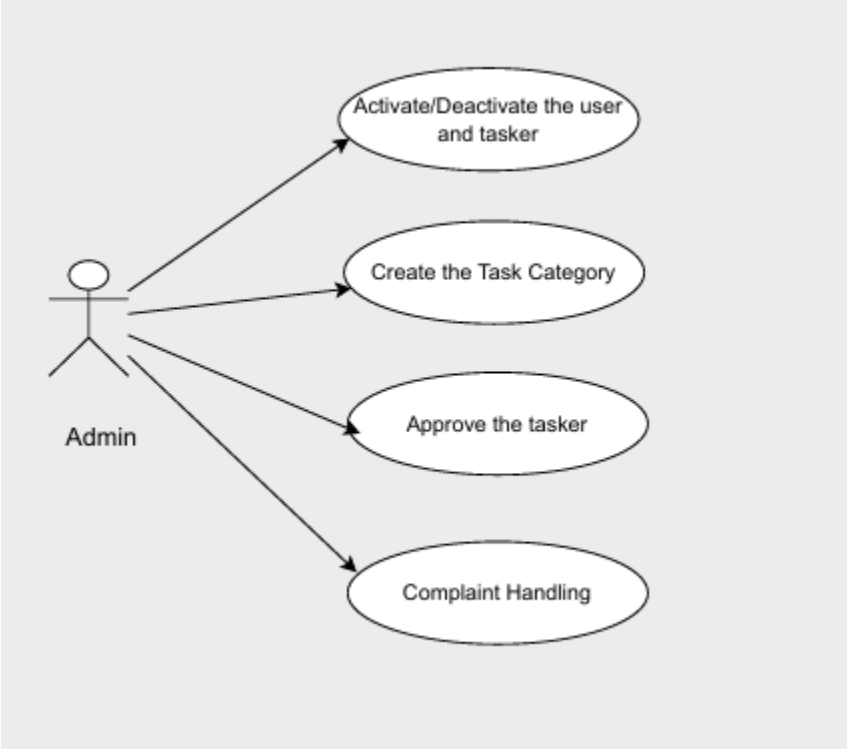
Customer:



Tasker:



Admin:


Activate/Deactivate the user and tasker
Create the Task Category
Approve the tasker
Complaint Handling
Admin

Non-Functional Requirement

3.2.1 Usability Requirement

The **usability requirement** ensures that the **TaskBuddy** platform is intuitive, user-friendly, and accessible to a diverse range of users, including **customers, taskers, and administrators**. The system must be designed to minimize complexity, reduce user errors, and provide a smooth experience across various devices and platforms.

1. **Ease of Navigation**
    - The interface should be simple and well-structured, allowing users to perform tasks such as **booking services, managing profiles, and processing payments** with minimal effort.
    - A **dashboard-based design** should be implemented to provide quick access to essential features.
2. **Mobile and Cross-Platform Compatibility**
    - The system must be **fully responsive** and accessible on desktops, tablets, and smartphones.
    - TaskBuddy should work across different web browsers (Chrome, Firefox, Edge, Safari) without UI or functionality issues.
3. **Minimal Learning Curve**
    - First-time users should be able to **register, search for services, and book taskers** easily without needing extensive guidance.
    - The platform should provide **on-screen tooltips, FAQs, and a help section** to assist users.
4. **Consistency in Design**
    - The UI should follow a **consistent theme, color scheme, and layout**, ensuring a uniform experience across different pages and features.
    - Icons and buttons should have **clear labels**, avoiding confusion among users.
5. **Error Prevention & User Feedback**
    - The system should provide **real-time validation** for form inputs (e.g., incorrect email format, weak passwords).

- Users should receive **instant feedback** on actions such as successful bookings, failed transactions, or incorrect inputs.

6. **Accessibility Compliance**
    - The platform should follow **WCAG (Web Content Accessibility Guidelines)** to ensure usability for people with disabilities.
    - Features like **keyboard navigation, screen reader support, and contrast adjustment** should be available for visually impaired users.

3.2.2 Performance Requirement

The **TaskBuddy** system must ensure **high performance, fast response times, and efficient resource utilization** to deliver a seamless user experience. Key operations such as **user authentication, tasker search, booking services, and payment processing** should be completed within **200-500 milliseconds** under normal conditions. The system should be capable of handling **multiple concurrent users** without noticeable delays, ensuring that customers and taskers can interact smoothly without performance degradation

Regular **performance testing, stress testing, and optimization** should be conducted to identify bottlenecks and improve efficiency. The platform should be optimized for **mobile and web access**, ensuring a smooth experience regardless of device or network conditions. By meeting these performance requirements, **TaskBuddy** will provide a **fast, scalable, and highly responsive** platform for both customers and taskers.

3.2.3 Reliability Requirement

To achieve reliability, **data redundancy and backup strategies** must be implemented. The system should use **cloud-based storage with automatic backups**, ensuring that user data, transactions, and bookings remain secure and recoverable in case of an outage. **Failover mechanisms** should be in place to redirect traffic to backup servers if the primary system encounters issues. Additionally, **error detection and logging** should be implemented to identify and resolve potential failures before they affect users.

The system must undergo **regular reliability testing**, including **stress testing, load testing, and failure recovery simulations**, to identify vulnerabilities and improve system stability. TaskBuddy should be able to handle **unexpected traffic surges and hardware failures** without disrupting user operations. By meeting these reliability requirements, TaskBuddy will offer a **trustworthy and consistent** experience, ensuring that customers and taskers can rely on the platform at all times.

3.2.4 Portability Requirement

The **TaskBuddy** system must be **highly portable**, ensuring that it can run seamlessly across different **devices, operating systems, and environments**. The platform should be accessible through **web browsers (Chrome, Firefox, Edge, Safari) and mobile devices (Android & iOS)** without requiring significant modifications. The system must be designed using **responsive web technologies** to ensure a consistent user experience across desktops, tablets, and smartphones.

To enhance portability, the backend should be **platform-independent**, allowing deployment on various cloud providers such as **AWS, Azure, or Google Cloud**. The application should also support **containerization (Docker/Kubernetes)** to enable easy migration between different hosting environments without compatibility issues. Additionally, the database should be compatible with multiple database management systems (Ms SQL) to provide flexibility in deployment.

The system must be **easily installable and configurable** across different environments, including **local development setups, testing servers, and production deployments**. Code dependencies should be minimized to reduce compatibility issues, and the software should adhere to **standard web protocols (RESTful APIs, JSON, HTTPS)** for smooth integration with third-party services. By meeting these portability requirements, **TaskBuddy** will remain **adaptable, scalable, and accessible** across diverse platforms, ensuring a seamless experience for all users.

3.2.5 Security Techniques

Security is a critical aspect of the **TaskBuddy** system, ensuring that **user data, transactions, and system resources** are protected against unauthorized access and cyber threats. The platform must implement **robust authentication mechanisms**, such as **JWT (JSON Web Token)-based authentication** or **OAuth**

To verify user identities and prevent unauthorized access. **Role-Based Access Control (RBAC)** should be enforced to restrict user privileges based on their roles (Customer, Tasker, Admin), ensuring that users can only access features relevant to their responsibilities.

To protect sensitive user information, the system must implement **data encryption techniques** both **in transit and AES-256 encryption for stored data**. **Secure password storage** using hashing algorithms like **bcrypt** should be employed to prevent credential leaks.

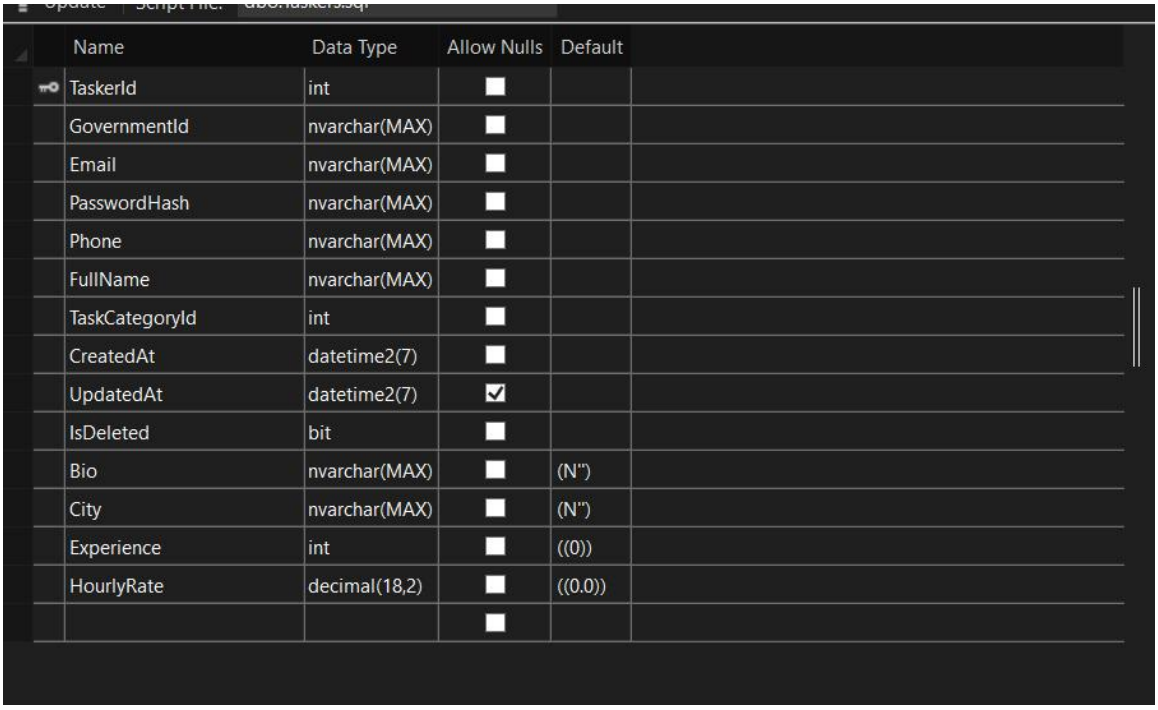
Additionally, **TaskBuddy** should implement **security best practices** such as **input validation, SQL injection prevention, and Cross-Site Scripting (XSS) protection** to safeguard against malicious attacks.

Secure session management techniques, such as **automatic session expiration and token revocation**, should also be implemented to protect user accounts. By incorporating these security techniques, **TaskBuddy** will ensure a **safe, trustworthy, and resilient** environment for both customers and taskers.

Database Design:

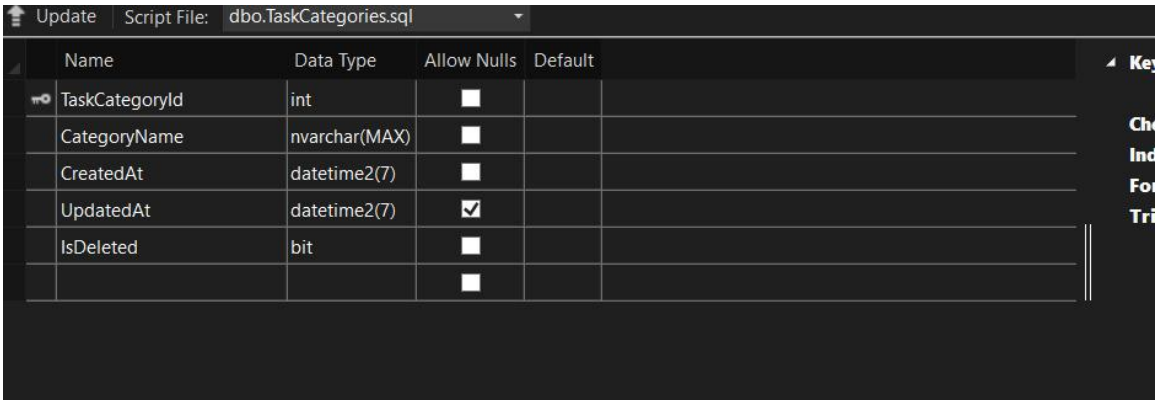
Column Name	Data Type	Constraints	Description
AdminId	INT	PRIMARY KEY, AUTO_INCREMENT	Unique identifier for each admin.
Name	VARCHAR(255)	NOT NULL	Stores the admin's full name.
Email	VARCHAR(255)	NOT NULL, UNIQUE	Admin's email (must be unique).
Password	VARCHAR(255)	NOT NULL	Stores the admin's hashed password.
CreatedAt	DATETIME	DEFAULT CURRENT_TIMESTAMP	Timestamp when the record is created.
UpdatedAt	DATETIME	NULL, DEFAULT CURRENT_TIMEST-AMP ONUPDATE CURRENT_TIMEST-AMP	Timestamp when the record is last updated .
IsDeleted	BOOLEAN	DEFAULT FALSE	Soft delete flag (false = active, true = deleted).

Customer Database:



Name	Data Type	Allow Nulls	Default
TaskerId	int	☐	
GovernmentId	nvarchar(MAX)	☐	
Email	nvarchar(MAX)	☐	
PasswordHash	nvarchar(MAX)	☐	
Phone	nvarchar(MAX)	☐	
FullName	nvarchar(MAX)	☐	
TaskCategoryId	int	☐	
CreatedAt	datetime2(7)	☐	
UpdatedAt	datetime2(7)	☑	
IsDeleted	bit	☐	
Bio	nvarchar(MAX)	☐	(N'')
City	nvarchar(MAX)	☐	(N'')
Experience	int	☐	((0))
HourlyRate	decimal(18,2)	☐	((0.0))
		☐	

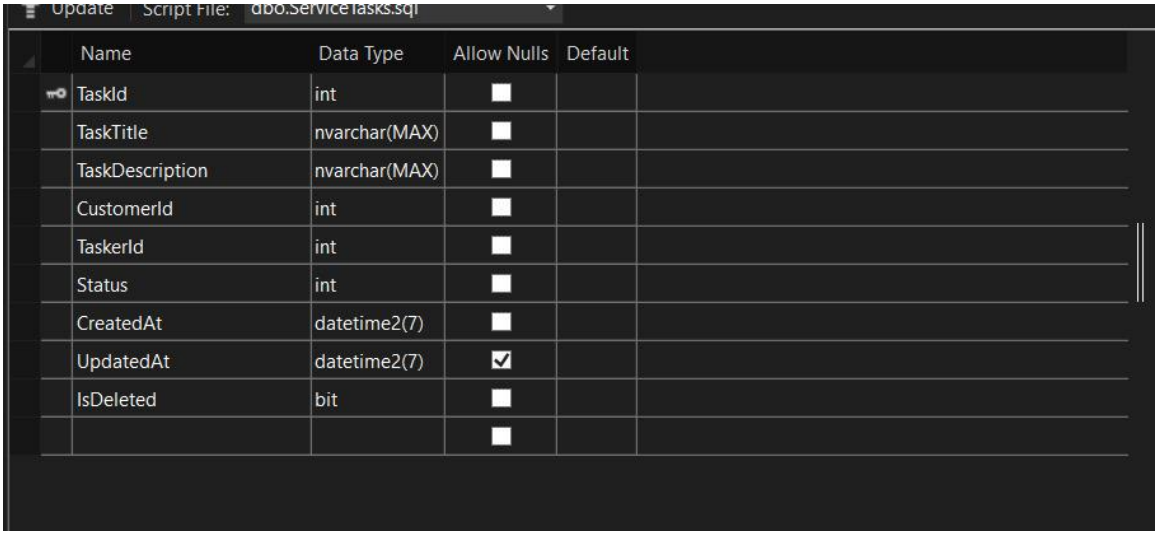
Task Category Database:



Update Script File: dbo.TaskCategories.sql

Name	Data Type	Allow Nulls	Default
TaskCategoryId	int	☐	
CategoryName	nvarchar(MAX)	☐	
CreatedAt	datetime2(7)	☐	
UpdatedAt	datetime2(7)	☑	
IsDeleted	bit	☐	
		☐	

Service Tasks:



Test Report

1. Test Objectives

- Validate that the system meets **functional requirements** such as **user registration, tasker booking, payments, and notifications**.
- Ensure compliance with **non-functional requirements** including **performance, usability, security, and reliability**.
- Identify and resolve any defects or inconsistencies before deployment.

2. Test Approach

- **Testing Methodology:** Manual and automated testing were performed.
- **Test Types:**
    - **Unit Testing** – To verify individual components like login, tasker search, and payment processing.
    - **Integration Testing** – To check data flow between different modules.
    - **System Testing** – To test the complete system functionality.
    - **Performance Testing** – To analyze response times and system load handling.
    - **Security Testing** – To check vulnerabilities like SQL injection, XSS, and authentication mechanisms.
    - **Usability Testing** – To evaluate the user experience across different devices.
- **Test Case Summary**

Test Case ID	Test Scenario	Expected Result	Actual Result	Status
TC-001	User Registration	New users should be able to register successfully	Success	Passed
TC-002	User Login	Valid users should log in securely	Success	Passed
TC-003	Tasker Booking	Customers should be able to book taskers	Success	Passed
TC-004	Payment Processing	Payments should be processed securely	Success	Passed

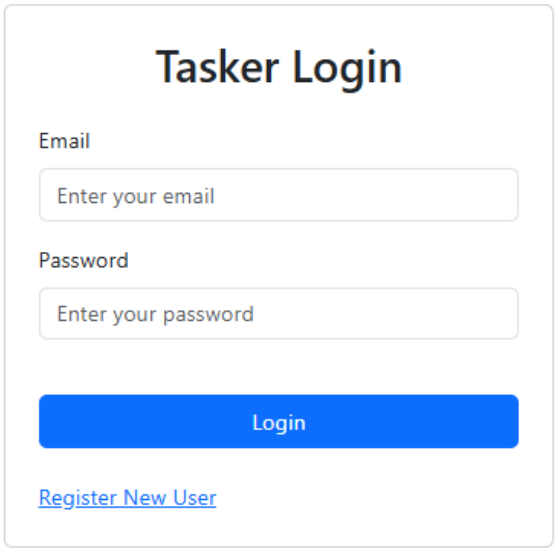
Test Case ID	Test Scenario	Expected Result	Actual Result	Status
TC-005	Load Handling	System should handle 1000+ concurrent users	No lag/errors	Passed
TC-006	Security (SQL Injection)	System should prevent SQL injection attacks	Blocked request	Passed

Test case report for Tasker:

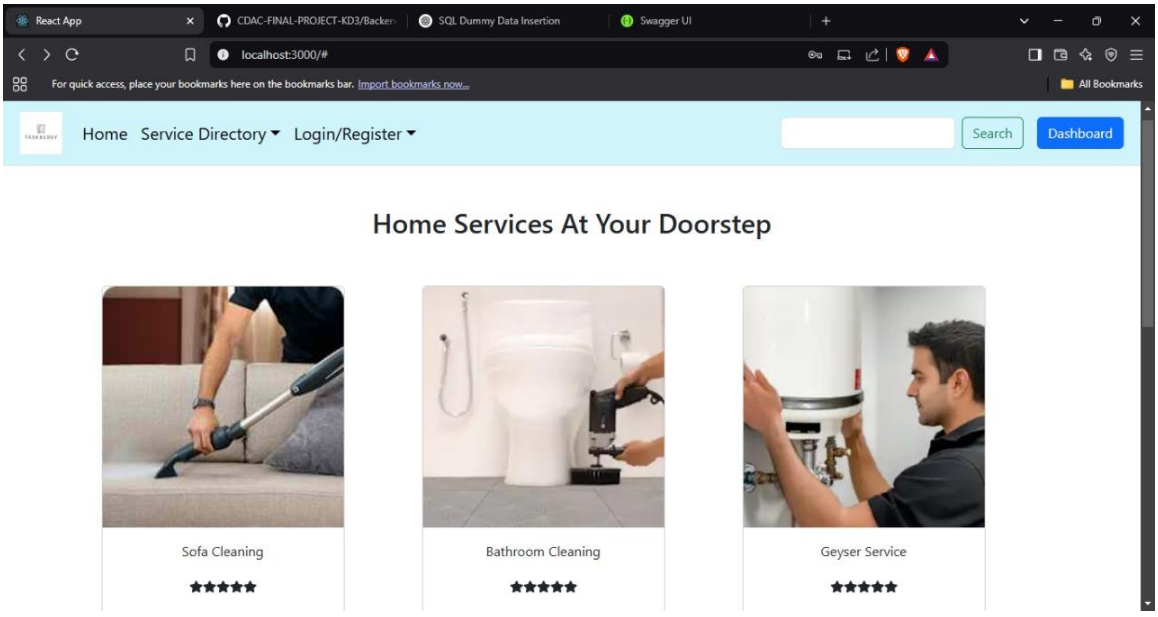
Test Case ID	Test Scenario	Expected Result	Actual Result	Status
TC-T001	Tasker Registration	Tasker should be able to register successfully with valid details	Success	Passed
TC-T002	Tasker Login	Tasker should log in securely with correct credentials	Success	Passed
TC-T003	Tasker Profile Update	Tasker should be able to update profile details (skills, availability, experience)	Success	Passed
TC-T005	Accept/Reject Booking	Tasker should be able to accept or decline a booking request	Success	Passed
TC-T006	Task Completion	Tasker should mark a booking as completed once done	Success	Passed
TC-T007	Payment Processing	Tasker should receive payments securely after task completion	Success	Passed
TC-T008	Tasker Rating & Review	Taskers should receive ratings and feedback from customers	Success	Passed

Appendix-B

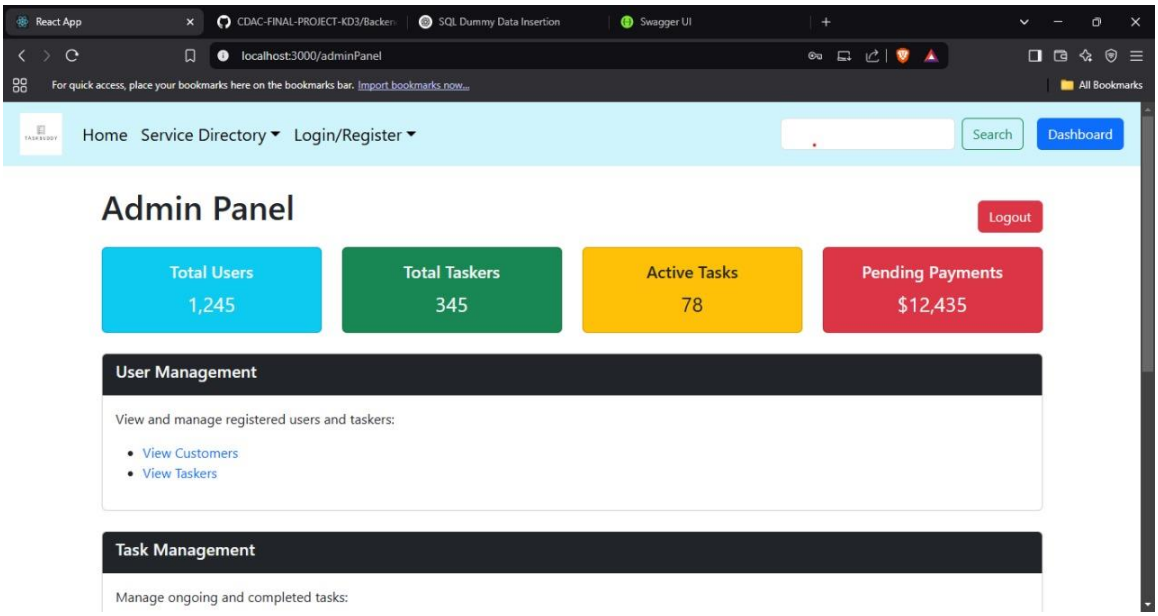
Login as tasker page:



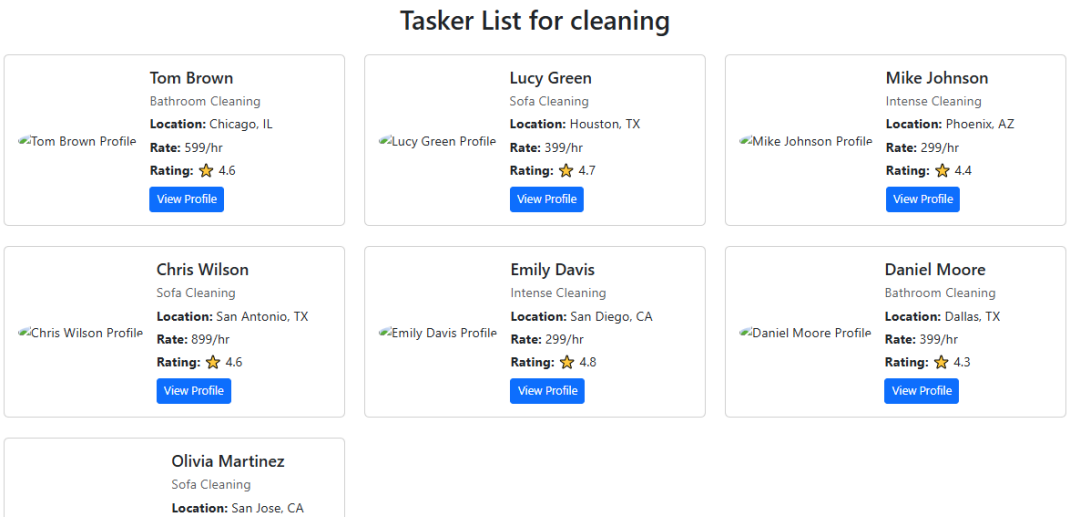
Home page:



Admin Panel:



Tasker profile page:



Conclusion

The **TaskBuddy** project successfully provides a seamless and efficient platform for connecting customers with skilled taskers for various household errands and specialized tasks. By integrating **user-friendly booking, secure payment processing, real-time notifications, and task management**, TaskBuddy ensures a smooth experience for both customers and taskers. The system's **functional and non-functional requirements** have been carefully designed and implemented to deliver reliability, usability, security, and high performance. With features such as **role-based access control, tasker ratings, and availability tracking**, the platform enhances trust and transparency between users.

Through extensive **testing and validation**, TaskBuddy has demonstrated its ability to handle real-world usage efficiently. The system successfully meets key objectives, including **tasker registration, service booking, secure payments, and performance optimization**. With its scalable architecture, **TaskBuddy** is well-equipped to support growing user demand. Future enhancements may include **AI-driven tasker recommendations, multilingual support, and mobile app integration** to further improve accessibility and convenience. In conclusion, TaskBuddy is a **robust, scalable, and secure solution** for simplifying task outsourcing, benefiting both customers and taskers.