

# **SKILLBRIDGE : A STUDENT TALENT MARKETPLACE PLATFORM**



**ANSILA NV**

**MES24MCA-2007**

**Department of Computer Applications**

**MES College of Engineering, Kuttippuram**

**20/08/2025**



PRODUCT OWNER

**Mr.BALACHANDRAN K P**

ASSOCIATE PROFESSOR

DEPARTMENT OF COMPUTER APPLICATIONS

MES COLLEGE OF ENGINEERING, KUTTIPPURAM



# TABLE OF CONTENTS

---

1. Introduction
2. Objective
3. Existing System
4. Proposed System
5. Motivation
6. Functionalities
7. Module Description
8. Developing Environment
9. Sprint Backlog
10. Product Backlog
11. User Story
12. Project Plans
13. Data Flow Diagrams
14. ER Diagram



# SKILLBRIDGE : A STUDENT TALENT MARKETPLACE PLATFORM

---

- Web-based platform that connects skilled students with clients (inside and outside campus) for freelance services such as graphic design, video editing, baking, drawing, tutoring, and more.
- Enables students to showcase their portfolio, register services with pricing/availability, and earn income, while clients can easily search, book, and hire students at affordable rates through a structured and verified system.



# SKILLBRIDGE : A STUDENT TALENT MARKETPLACE PLATFORM

---

- Admin dashboard provides full control over student and client accounts, service verification, booking oversight, report generation, and system monitoring.
- Built using HTML, CSS, JavaScript on the frontend and Python (Django) on the backend with MySQL/Firebase as the database. The platform ensures transparency, provides in-app notifications, and builds a mini-campus economy that fosters student entrepreneurship and financial independence.



# OBJECTIVES

---

- Help students earn money by showing their skills online.
- Allow clients to easily find and book student services.
- Give students a chance to build experience and portfolios.
- Let admins manage users, services, and reports in one place.
- Create a safe and trusted platform for both students and clients.
- Build a simple, modern, and scalable system.
- Provide notifications, messages, and reviews for better communication



# EXISTING SYSTEM

---

- Students use word-of-mouth to share skills.
- No single platform to find or hire students.
- Skills are not visible to many people.
- No trust or verification system.
- Many students lose chances to earn and gain experience.
- No tracking of bookings or payments.

# PROPOSED SYSTEM

---

- A digital platform for students to list and showcase their skills.
- Clients can easily search, book, and hire students.
- Students get more visibility and earning opportunities.
- Secure system with admin verification and reviews.
- Improved opportunities for students and satisfaction for clients.
- Tracks services, bookings, and payments in one place.





# MOTIVATIONS

---

- Many students have valuable skills but no platform to showcase them.
  - To increase visibility of student talents beyond campus boundaries.
  - To create earning opportunities for students during their academic life.
  - To help clients easily discover affordable and skilled student services.
  - To ensure trust through verified profiles, ratings, and reviews.
  - Encourages student entrepreneurship and financial independence.
- 



# FUNCTIONALITIES

---

- Students can create profiles and list their skills.
- Clients can search and book student services..
- Easy booking, rescheduling, and cancellation.
- Notifications for bookings, updates, and payments.
- Ratings and reviews after each service..

# FUNCTIONALITIES

---

- Students can track their earnings and booking history.
- Clients can view past bookings and payments.
- Admin can manage users and verify services.
- Reports on services, bookings, and student earnings.
- Simple, secure, and user-friendly system.

# MODULE DESCRIPTION

---

## 1. Admin Module

- Login
- Verify service
- View users
- View ratings
- View complaint & send replay
- Block / Unblock
- User / Service

# MODULE DESCRIPTION

---

## 2. Service Module

- Registration
- Login
- Manage Works
- View Request from User
- Chat
- Update Status
- Update Price
- View feedback & rating
- Send complaint
- View replay

# MODULE DESCRIPTION

---

## **3. User Module**

- Registration
- Login
- Search products
- Request for products
- Chat
- View status
- Payment
- Feedback & Rating
- Send complaint
- View replay

# DEVELOPING ENVIRONMENT

---

- **Operating System:** Windows / Linux
- **Front End:** HTML,CSS,JavaScript(Responsive Design)
- **Back End:** Django (Python)
- **Database:** MySQL / Firebase (Relational / Cloud Database)
- **Authentication:** Role-based with Secure Login
- **Location Services:** Integrated Location for Nearby Search
- **IDE:** PyCharm
- **Version Control:** Git & GitHub



# SPRINT BACKLOG

Backlog tem	Status And Completio n Date	Original Estimatio n in Hours	Day 1 hrs	Day 2 hrs	Day 3 hrs	Day 4 hrs	Day 5 hrs	Day 6 hrs	Day 7 hrs	Day 8 hrs	Day 9 hrs	Day 10 hrs
SPRINT 1												
Project setup	06/08/2025	3	1	1	1	0	0	0	0	0	0	0
Database Schema Design	10/08/2025	5	1	1	1	1	1	0	0	0	0	0
Module Planning & Routing	12/08/2025	4	1	1	1	1	0	0	0	0	0	0
SPRINT 2												
Admin Login & Dashboard	15/08/2025	6	1	1	1	1	1	1	0	0	0	0
Service & User Verification	18/08/2025	5	1	1	1	1	1	0	0	0	0	0



# SPRINT BACKLOG

Backlog tem	Status And Completion Date	Original Estimatio n in Hours	Day 1 hrs	Day 2 hrs	Day 3 hrs	Day 4 hrs	Day 5 hrs	Day 6 hrs	Day 7 hrs	Day 8 hrs	Day 9 hrs	Day 10 hrs
SPRINT3												
Student Registratio n & Portfolio	27/08/2025	6	1	1	1	1	1	1	0	0	0	0
Request Managemen t & Pricing	01/09/2025	7	1	1	1	1	1	1	1	0	0	0
SPRINT 4												
Client Registratio n	13/10/2025	6	1	1	1	1	1	1	0	0	0	0
Feedback & Complaint Handling	08/10/2025	7	1	1	1	1	1	1	1	0	0	0
TOTAL		50	9	9	8	6	6	6	6			



# PRODUCT BACKLOG

ID	NAME	PRIORITY <high/medium/low>	ESTIMATE (Hours)	STATUS <Planned/In progress/Completed>
1	Admin Dashboard & Login	High	3	completed
2	Service Verification & Approval	High	8	completed
3	User Management (Block/Unblock)	Medium	6	completed
4	Complaint/Issue Resolution Interface	High	10	completed
5	View service ratings/quality monitoring	Medium	5	completed

# PRODUCT BACKLOG

ID	NAME	PRIORITY <high/medium/low>	ESTIMATE (Hours)	STATUS <Planned/In progress/Completed>
6	Service provider Registration & Login	High	5	completed
7	Service management	High	12	completed
8	View client Requests &Details	High	7	completed
9	Update Request status	High	5	completed
10	Chat with client	High	15	completed

# PRODUCT BACKLOG

ID	NAME	PRIORITY <high/medium/low>	ESTIMATE (Hours)	STATUS <Planned/In progress/Completed>
11	Client Registration & Login	High	5	completed
12	Browse/search services	High	10	completed
13	Send service request	High	5	completed
14	Chat with Service provider	High	15	completed
15	Track Request & secure payment	High	18	completed

# USER STORY

---

User Story ID	As a type of User	I want to <Perform some task>	So that i can <Achieve Some Goal>
1	ADMIN	Login	Access the admin dashboard securely
2	ADMIN	Manage Users & Services	Approve, verify, or block student/client accounts and services
3	ADMIN	Monitor Bookings	Oversee service requests and resolve disputes
4	ADMIN	Generate Reports	Track student earnings, service demand, and client activity

# USER STORY

---

User Story ID	As a type of User	I want to <Perform some task>	So that i can <Achieve Some Goal>
5	ADMIN	Manage Reviews	Remove inappropriate ratings/reviews for system trustworthiness
6	USER(Student)	Register & Login	Create an account and access my profile
7	USER(Student)	Create Portfolio	Showcase my skills and previous work
8	USER(Student)	Create Service Listing	Add services with price, category, and availability

# USER STORY

---

User Story ID	As a type of User	I want to <Perform some task>	So that i can <Achieve Some Goal>
9	USER(Student)	Manage Earnings	Track income from completed services
10	USER(Client)	Search Services	Find students based on skills, categories, or location
11	USER(Client)	Book Service	Hire a student for a project or task
12	USER (Client)	Provide Feedback	Share ratings and reviews after service completion

# USER STORY

<b>User Story ID</b>	<b>As a type of User</b>	<b>I want to</b> <Perform some task>	<b>So that i can</b> <Achieve Some Goal>
13	USER (Student & Client)	Messaging	Communicate directly for clarifications and updates
14	USER (Student & Client)	Receive Notifications	Get booking confirmations, status updates, and announcements
15	USER (Student)	Access Discount System	Offer/receive discounted rates for internal campus users



# PROJECT PLAN

User StoryID	Task Name	Start Date	End Date	Days	Status
1	SPRINT 1	06/08/25	08/08/25	19	In Progress
2		09/08/25	15/08/25		In Progress
3		16/08/25	24/08/25		In Progress
4,5	SPRINT 2	25/08/25	02/09/25	18	In Progress
10,11		03/09/25	11/09/25		In Progress

# PROJECT PLAN

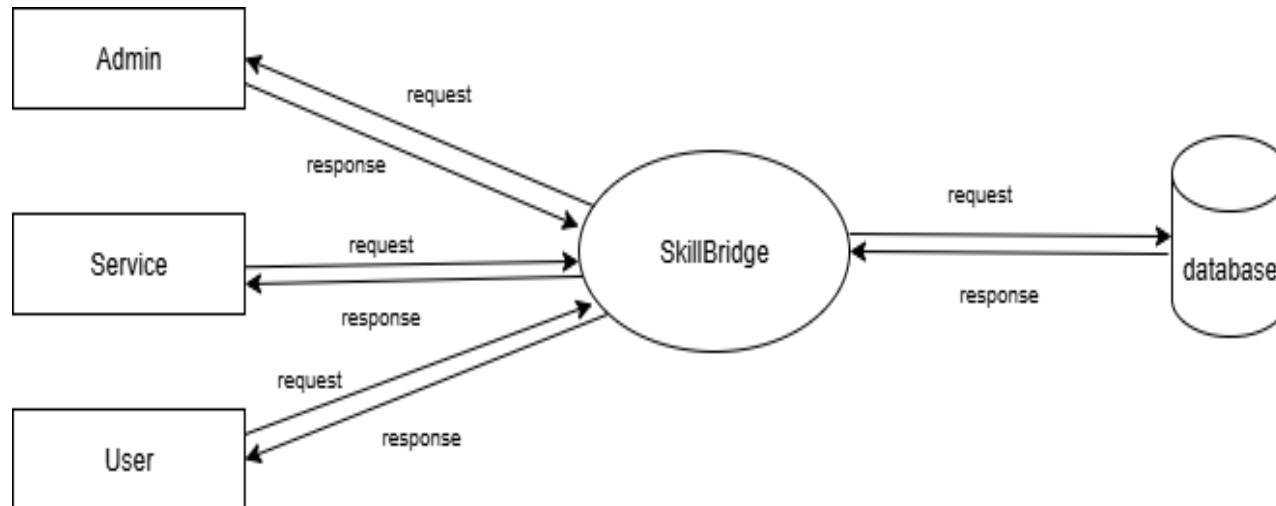
---

User StoryID	Task Name	Start Date	End Date	Days	Status
6,7	SPRINT 3	12/09/25	20/09/25	18	In Progress
12,13		21/09/25	29/09/25		In Progress
8,9	SPRINT 4	30/09/25	08/10/25	13	In Progress
14,15		09/10/25	12/10/25		In Progress

# DATA FLOW DIAGRAM

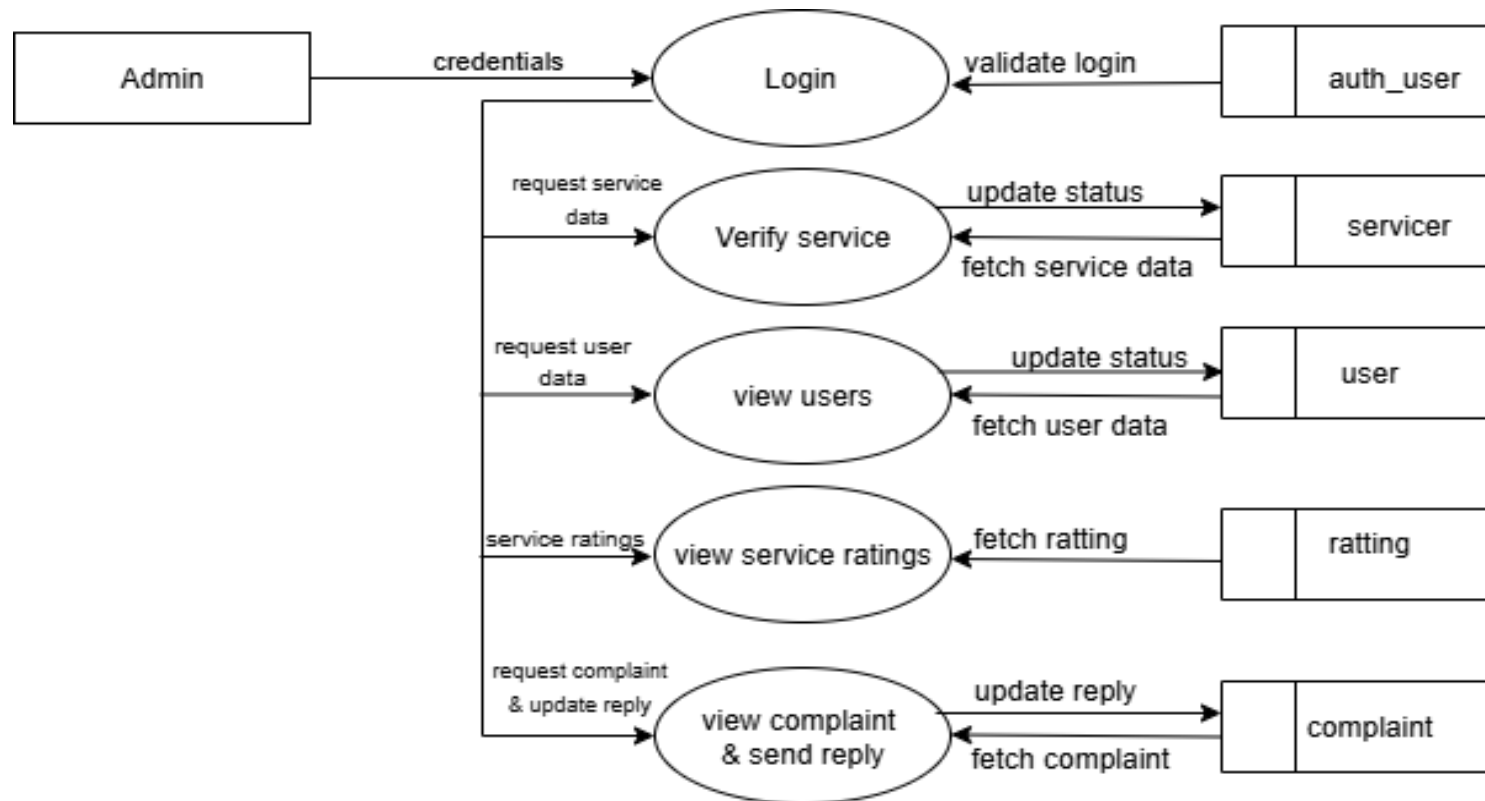
---

- LEVEL 0



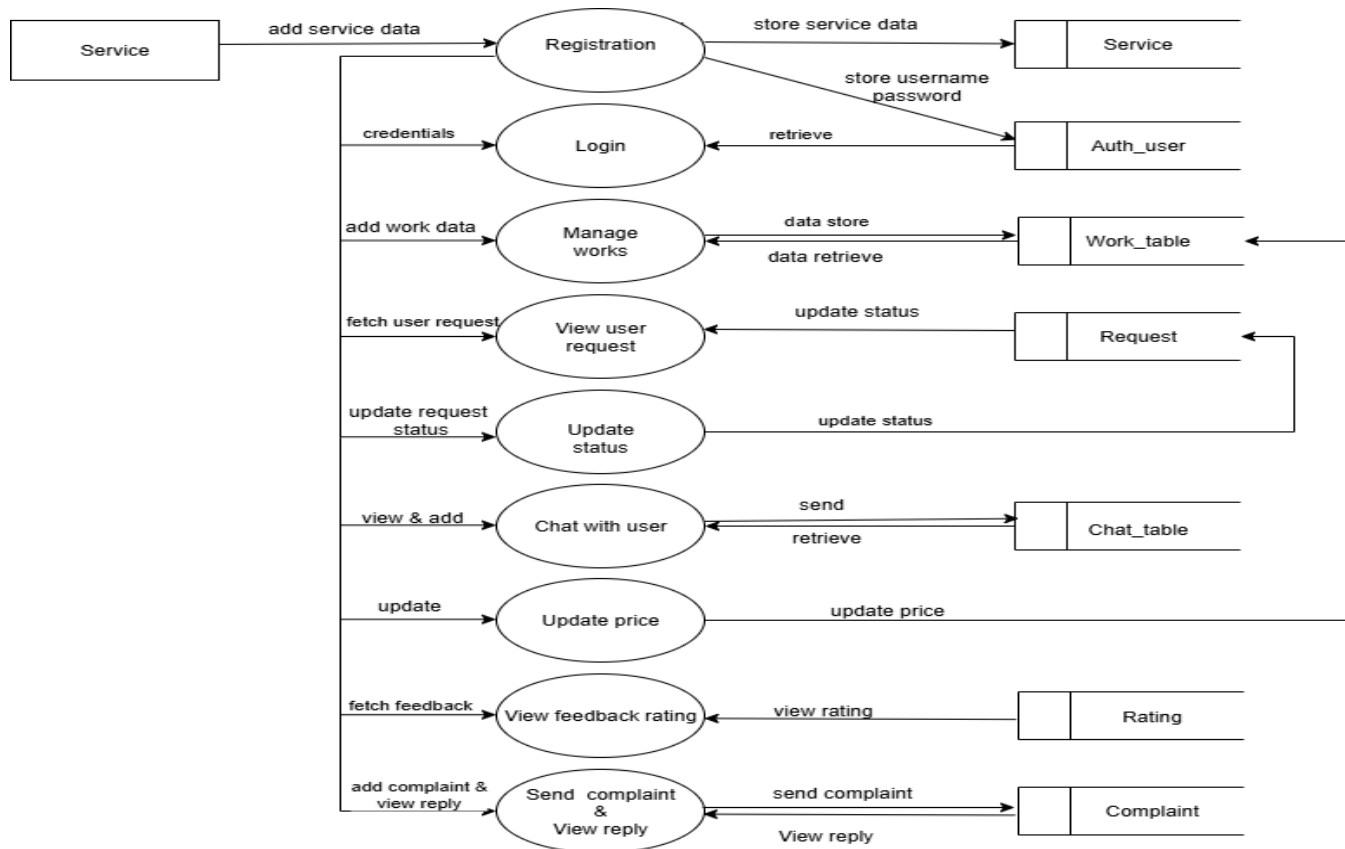
# DATA FLOW DIAGRAM

- Level 1



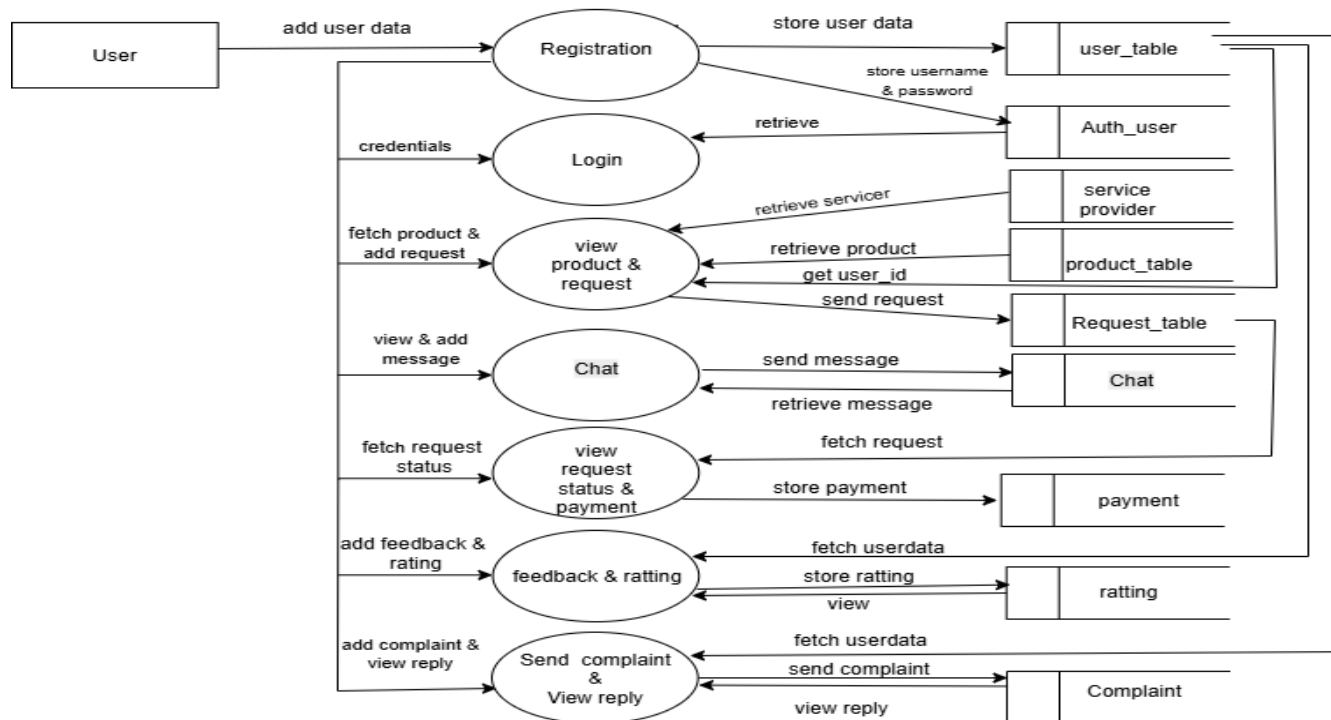
# DATA FLOW DIAGRAM

- Level 2

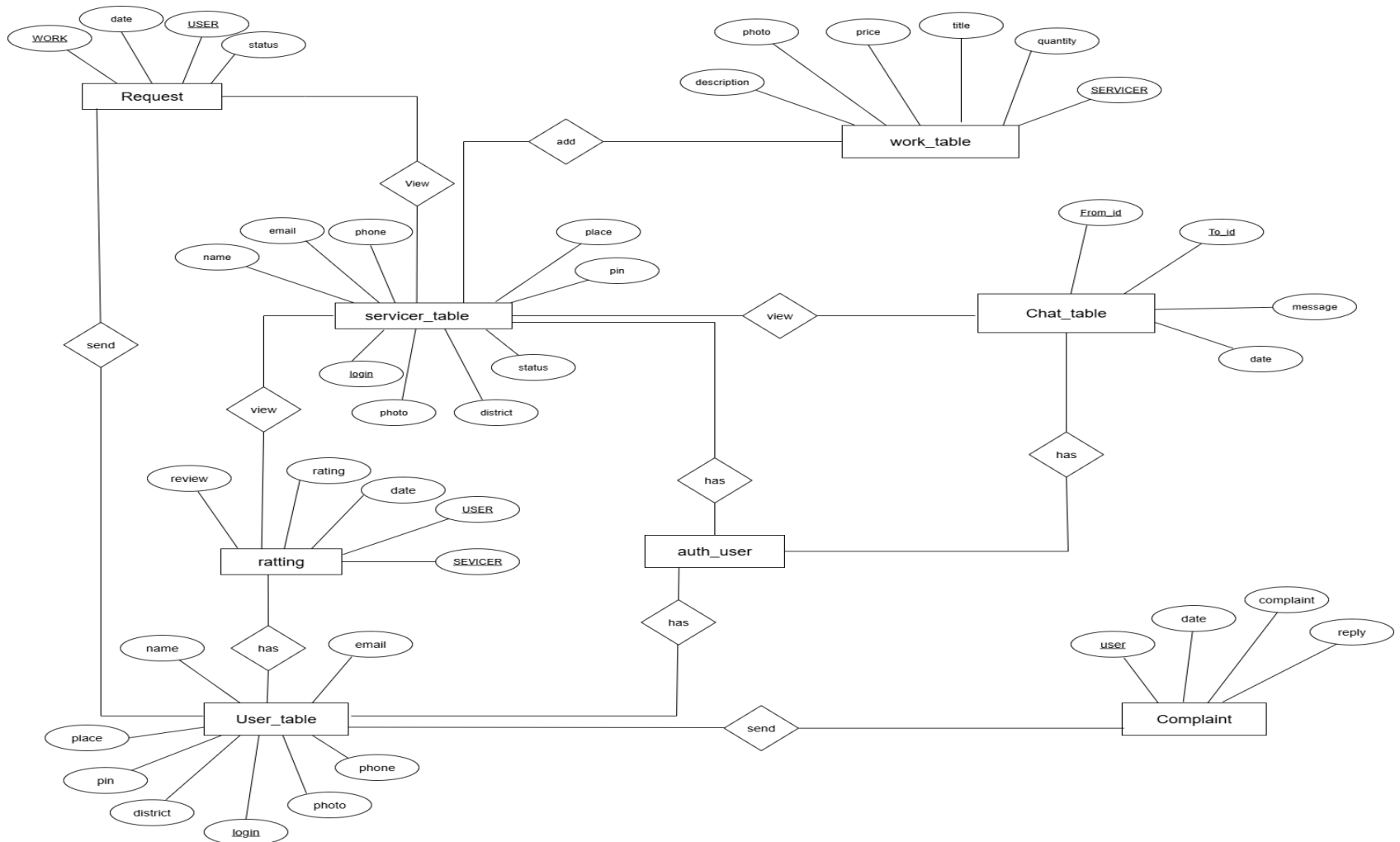


# DATA FLOW DIAGRAM

- Level 3

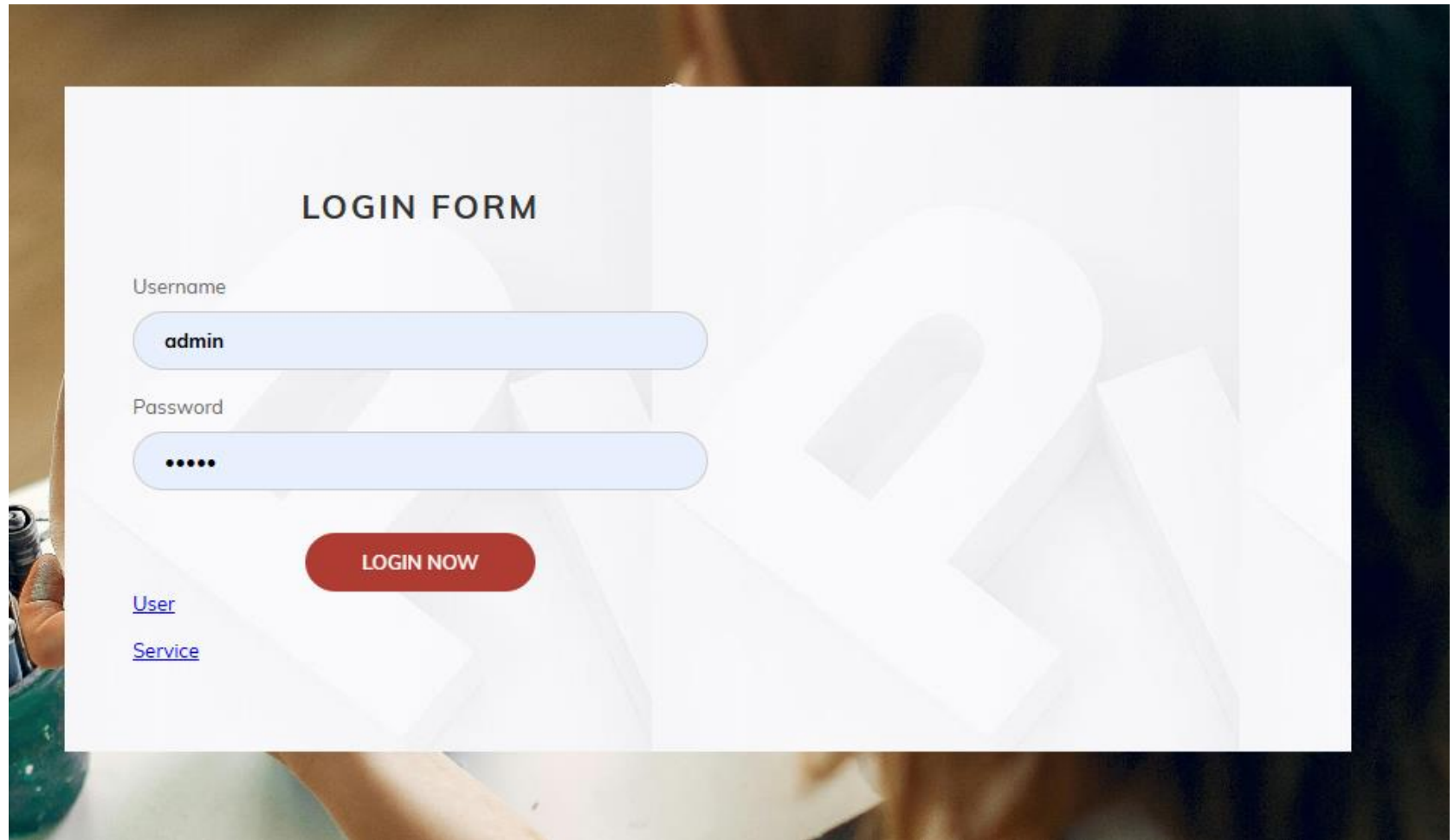


# ER DIAGRAM



# LOGIN

---



**LOGIN FORM**

Username

**admin**

Password

.....

**LOGIN NOW**

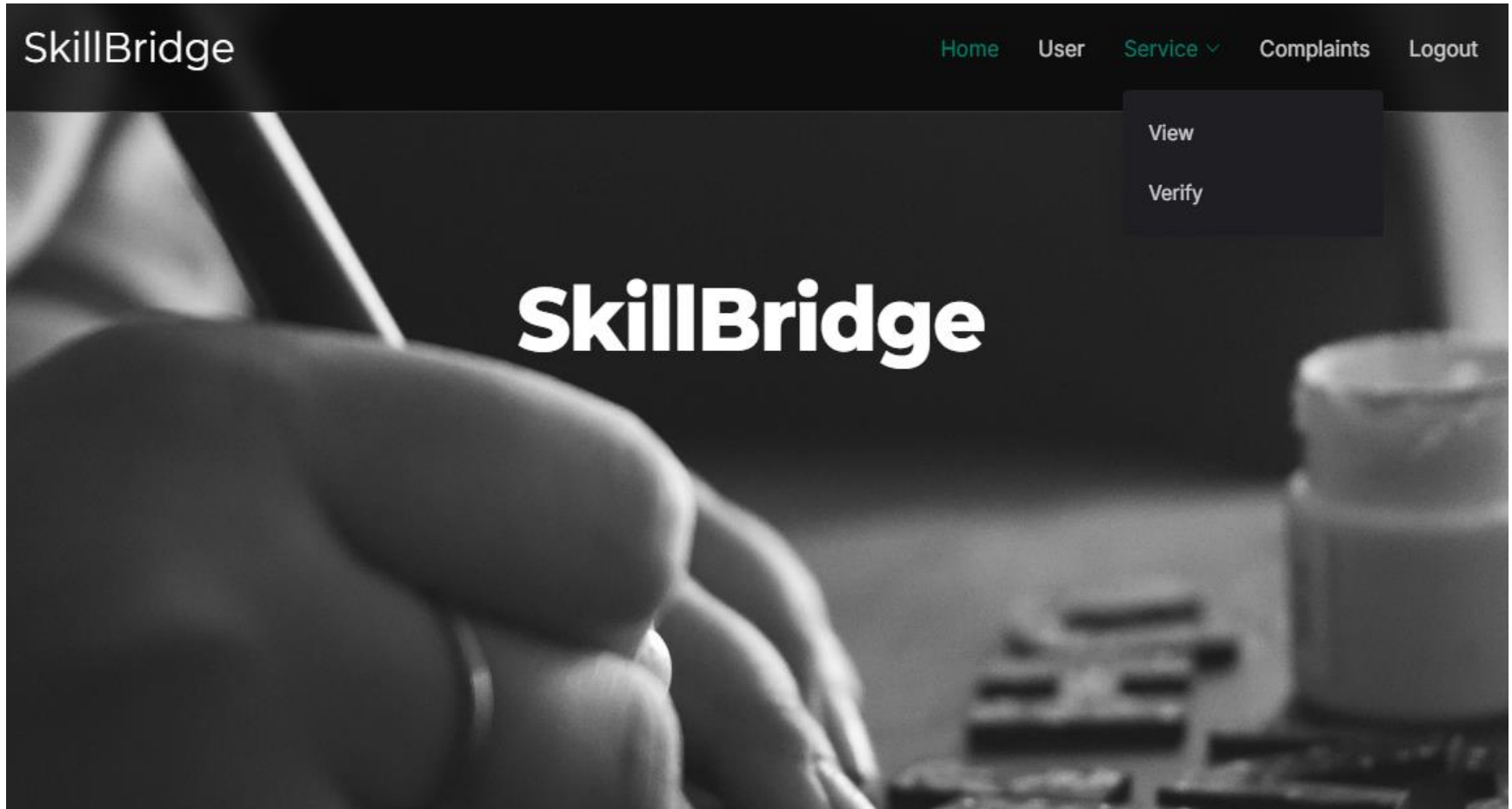
[User](#)

[Service](#)



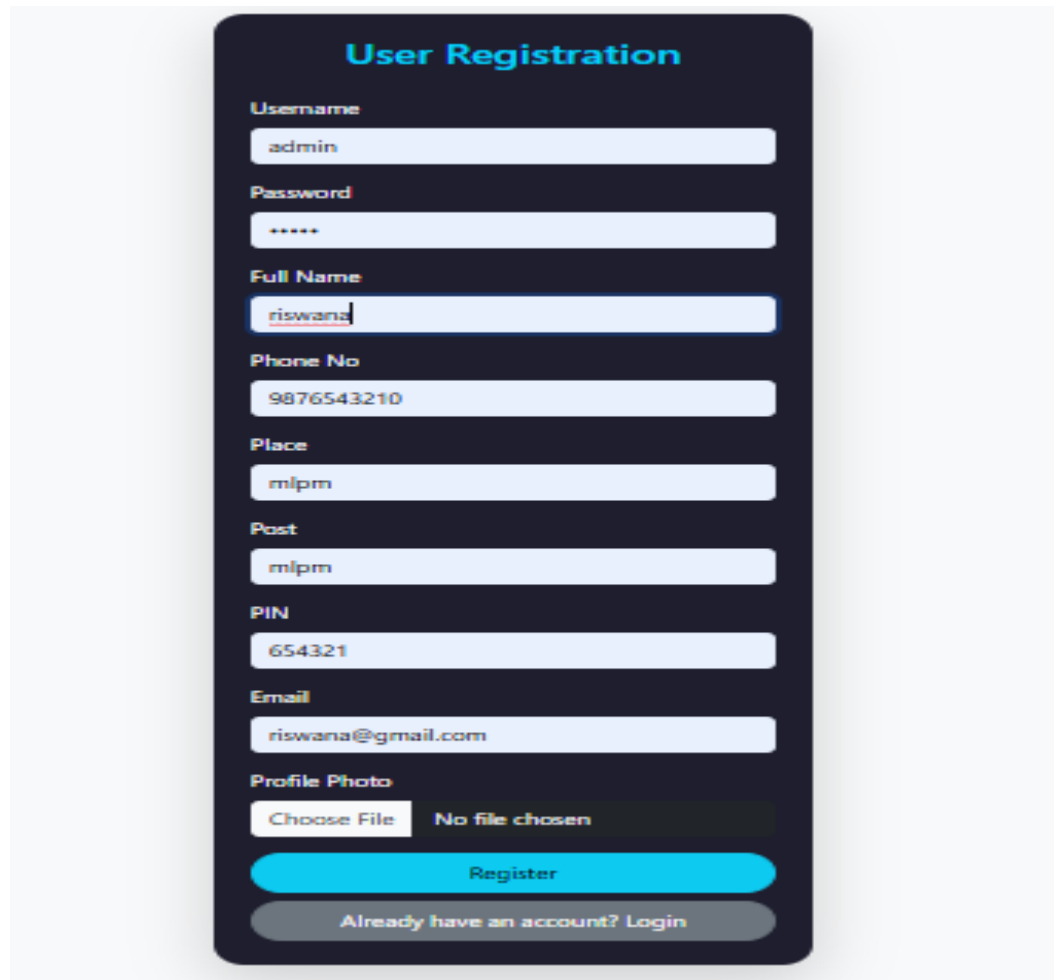
# INTERFACE

---



# INTERFACE

---



The image shows a user registration form with a dark blue background and light blue text. The form is titled "User Registration" in a bold, light blue font. It contains several input fields for registration details, each with a label above it. The fields are: Username (admin), Password (masked with dots), Full Name (riswana), Phone No (9876543210), Place (mlpm), Post (mlpm), PIN (654321), and Email (riswana@gmail.com). There is also a "Profile Photo" section with a "Choose File" button and a "No file chosen" status. At the bottom, there is a large blue "Register" button and a grey "Already have an account? Login" button.

**User Registration**

Username  
admin

Password  
\*\*\*\*\*

Full Name  
riswana

Phone No  
9876543210

Place  
mlpm

Post  
mlpm

PIN  
654321

Email  
riswana@gmail.com

Profile Photo  
Choose File No file chosen

Register

Already have an account? Login

# INTERFACE

---

### Service Registration

Name

Phone No

Place

Post

PIN Code

Email

Photo

Username

Password

Your registration will be marked as pending until approved.

# FUNCTIONALITIES

```
def loginpost(request):
    username = request.POST["username"]
    password = request.POST["password"]

    user = authenticate(request, username=username, password=password)
    print(user)
    if user is not None:
        print("hhh")
        if user.groups.filter(name="Admin").exists():
            print("jajaja")
            login(request, user)
            return redirect('/myapp/admin_home/')
        if user.groups.filter(name="Service").exists():
            print("jajaja")
            ob=Service_Table.objects.get(LOGIN__id=user.id)
            if ob.status=='service':
                login(request, user)
                return redirect('/myapp/service_home/')
        if user.groups.filter(name="User").exists():
            print("jajaja")
            login(request, user)
            return redirect('/myapp/user_home/')
    else:
        messages.warning(request, "Invalid username or password")
        return redirect('/myapp/')
    messages.warning(request, "Invalid username or password")
    return redirect('/myapp/')
```

# FUNCTIONALITIES

---

```
def registration_service(request):
    if request.method == "POST":
        name = request.POST["name"]
        phoneno = request.POST["phoneno"]
        place = request.POST["place"]
        post = request.POST["post"]
        pin = request.POST["pin"]
        email = request.POST["email"]
        password = request.POST["password"]
        username=request.POST['username']
        photo = request.FILES["photo"]
        user = User.objects.create(username=username, password=make_password(password), email=email, first_name=name)
        user.save()
        user.groups.add(Group.objects.get(name="Service"))
        Service_Table.objects.create(
            LOGIN=user,
            name=name,
            phoneno=phoneno,
            place=place,
            post=post,
            pin=pin,
            email=email,
            photo=photo,
        )
        messages.success(request,"Success")
        return redirect("/myapp/")
```

# FUNCTIONALITIES

```
@login_required(login_url='/myapp/')
def service_home(request):
    return render(request, "serviceindex.html")

@login_required(login_url='/myapp/')
def view_profile(request):
    profile=Service_Table.objects.get(LOGIN__id=request.user.id)
    return render(request, "service/profile.html", {"profile":profile})

@login_required(login_url='/myapp/')
def update_profile(request):
    profile=Service_Table.objects.get(LOGIN__id=request.user.id)
    if request.method == "POST":

        profile.name = request.POST["name"]
        profile.phoneno = request.POST["phoneno"]
        profile.place = request.POST["place"]
        profile.post = request.POST["post"]
        profile.pin = request.POST["pin"]
        profile.email = request.POST["email"]

        if "photo" in request.FILES:
            profile.photo = request.FILES["photo"]

    profile.save()
```

# FUNCTIONALITIES

```
@login_required(login_url='/myapp/')
def view_works(request):

    works = Work_Table.objects.filter(SERVICE__LOGIN__id=request.user.id)
    return render(request, "service/mywork.html", {"works": works})

@login_required(login_url='/myapp/')
def add_work(request):
    service = Service_Table.objects.get(LOGIN=request.user)

    if request.method == "POST":
        title = request.POST.get("title")
        description = request.POST.get("description")
        price = request.POST.get("price")
        photo = request.FILES.get("photo")

        work = Work_Table.objects.create(
            SERVICE=service,
            title=title,
            description=description,
            price=price,
            photo=photo,
            date=datetime.today()
        )
        return redirect("/myapp/view_works/#about")

    return render(request, "service/addwork.html")
```

# CONCLUSION AND FUTURE SCOPE

---

The SkillBridge: A Student Talent Marketplace Platform effectively addresses the challenges faced by students in showcasing and monetizing their skills. By offering a centralized and structured platform, it enables students to connect with potential clients within and outside their campus in a secure and organized manner. The system ensures smooth communication, safe payments, and transparency through features like real-time chat, request tracking, and admin verification. SkillBridge not only helps students gain practical experience and income but also builds client trust by ensuring the quality and authenticity of services. Overall, the platform successfully achieves its goal of empowering student freelancers and simplifying service access for clients.





# CONCLUSION AND FUTURE SCOPE

---

There is significant potential to enhance the SkillBridge platform in future versions. A dedicated mobile application can be developed to make the platform more accessible and user-friendly. AI-based matching between clients and service providers can improve efficiency and user experience. Features like automated review moderation, digital certification for verified talents, and support for multiple payment gateways can increase platform credibility. Integration with university portals, internship listings, or even placement services can also expand the platform's reach and utility. With ongoing improvements, SkillBridge can evolve into a comprehensive career-launching ecosystem for students



THANK YOU