FINAL YEAR PROJECT

By

Eman Noor **057680**
Tahreem Azeem **057722**

Under the supervision of
Miss Sadaf Sikandar

Bachelor of Science in INFORMATION TECHNOLOGY (2021-2025)
FACULTY OF COMPUTING & INFORMATION TECHNOLOGY (FCIT),
UNIVERSITY OF THE PUNJAB, LAHORE.

A project presented to
University of the Punjab, Lahore

In partial fulfillment
of the requirement for the degree of

Bachelors of Science in INFORMATION TECHNOLOGY (2021-2025)

By

Tahreem Azeem	**057722**
Eman Noor	**057680**

FACULTY OF COMPUTING & INFORMATION TECHNOLOGY (FCIT),
UNIVERSITY OF THE PUNJAB, LAHORE.

DECLARATION

We declare that this software and report are completely our own work. Nothing has been copied from any sources. If any part s later found to be copied or taken from somewhere else, we accept full responsibility. This work has not been submitted for any other degree or course at any other institute.

Signature: ---------- **Signature: ----------**

Tahreem Azeem Roll no. 057722 **Eman Noor Roll no. 057680**

CERTIFICATE OF APPROVAL

It is to certify that the final year design project (FYDP) of BS (INFORMATION TECHNOLOGY) **"Barter Brains"** is developed by **Tahreem Azeem [057722]** and **Eman Noor [057680]** under the supervision of **"MS SADAF SIKANDAR"** in my opinion; it is fully adequate in scope and quality for the degree of Bachelors of Science In Information Technology

Signature: ----------

FYDP Supervisor: Ms Sadaf Sikandar

Signatures (Faculty Advisory Committee (FAC)

Signatures			
Name			
	FAC1	FAC2	FAC3

Signature: ----------

Head of FYDP Coordination Office:

Signature: ---------- Dated: ----------

Chairperson, Dept of Information Technology

Executive Summary

Barter Brains is a web-based platform created to make it easy for people to share and exchange skills. It helps individuals teach what they know and learn new things from others all free of cost. The goal is to build a supportive community where users can grow together by exchanging knowledge and talents. After successful skill-sharing, users are awarded a digital certificate**.**

Objectives:

- Offer their skills to teach others.
- Find and learn skills they are interested in.
- Connect with people for mutual skill exchange.
- Receive digital certificates after completing a skill exchange.

Methodology:

Barter Brains was developed using:

- **Frontend:** HTML, CSS (with Bootstrap), and JavaScript
- **Backend:** PHP (Laravel Framework)
- **Database:** MySQL

The platform follows the **MVC (Model-View-Controller)** architecture for clean, modular design and easy maintenance.

Conclusion:

Barter Brains is a functional, user-friendly platform that makes skill exchange easy and free. It creates a community where people can learn, teach, and grow together. With its clear design and automated certificate generation, the platform encourages active participation and promotes lifelong learning.

Keywords:

- Skill Exchange Platform
- Laravel Web Application
- Digital Certification
- Peer-to-Peer Teaching

Acknowledgement

We would like to express our heartfelt thanks to **Govt. Graduate College For Women, Gujranwala** for providing us with the opportunity and support to complete on FINAL YEAR PROJECT, ***Barter Brains***. We are especially thankful to our respected supervisor, **Ms. Sadaf Sikandar**, for her continuous guidance, encouragement, and valuable feedback throughout the project. Her support helped us stay focused and motivated. Lastly, we are deeply grateful to our families and friends for their love, patience, and constant support during this journey. Without them, this project would not have been possible.

Signature: ---------- **Signature: ----------**

Tahreem Azeem **Roll no. 057722** **Eman Noor** **Roll no. 057680**

Table of Contents

Chapter 1

Introduction

1. Introduction

Across Barter Brains, skill-exchange platform where people swap skills, like trading coding lessons for music classes. Allowing for collaborative learning, it helps users learn new skills without spending money. The platform includes skill categories, user profiles, and requests for exchanges, facilitates learning and development.

1.1 Problem Statement

In today's busy world, many people want to learn new skills but can't afford expensive classes or find the right tutors. At the same time, people have valuable skills they could teach others. However, there is no common and trusted platform where people can connect and exchange skills without money.
Barter Brains is being developed to solve this problem by providing a system where users can offer a skill they know and request a skill they want to learn. In example, a person who knows graphic designing can teach it in exchange of learning coding from someone else. This system promotes learning through collaboration and builds a skill-sharing community.

1.2 Problem Solution

Barter Brains is a web-based platform that connects people who wants to swap their skills. Barter Brains allows users to create a profile, search for matching users, send skill swap request, and schedule sessions. Barter Brains, through this software solution, makes learning a two-way street where everyone is both a teacher and a learner.
Some main facilities of Barter Brains:

To create a user-friendly platform for skill swap

To match users based on their skills and interest

To provide learning without financial cost

To support personal growth and community development

To build trust through users reviews

To track learning history and exchange sessions

To make the plate form accessible on mobile and desktop

To notify users about skill swap request

To support multiple categories of skills like art, tech, music, cooking etc

1.3 Objectives of the Proposed System

The primary objective of the Barter Brains system is to enable individuals to barter skills by having them serve as learners and teachers. The system exists to facilitate peer-to-peer learning in which user can teach a skill and learn another skill from another user.

To measure the success, the platform will track how many skills exchange happens each week, how satisfied the user are on the basis of their reviews or feedback and also how quick multiple user being matched for skills exchange.

The system is designed to be achievable for everyone. It is also relevant in today's world where people are interested in learning from one another building communities without spending a lot of money.

The system focuses on what user want and need it will have an easy to use deign and will be improved on the basis of feedback who test it early.

This platform avoids unclear or confusing languages or words. Al things will be explained I an easy language so the user will know how to use it without facing any difficulty. The developing team will also not set unrealistic targets and make sure everything is possible according to the time and resources available.

1.4 Scope

Barter Brains enables users to connect with each other for skill exchange. Users will be able to build their skill profile, list the skills they can teach and those they want to learn and match with other users based on shared interests. The platform will allows users to send and receive exchange request, schedule sessions and give review after each interaction.
The system will not include online payments or built-in video calling features, and it will not verify the skill levels of users, reviews will handle that aspect. The main stakeholders are skill seekers, skill providers, administrators, and developers of the project.

The scope fully aligns with the project's goal of creating a free and accessible platform for exchanging knowledge. High-priority features of Barter Brains include user registration, skill search and match, and scheduling options. Key milestones of Barter Brains include requirement gathering, UI/UX design, backend setup, feature development, testing, and final deployment.

1.5 System Components

The Barter Brains system consists of several modules and each module handles specific tasks that together support the complete skill exchange process.

1.5.1 Client Web Module

These modules are designed for users who access the system through a desktop or laptop browser.

Module 1: User Account Management

This module helps users create and manage their accounts on the platform.

Sign Up/Login: Users can create an account with their email or phone number and log in securely.

Profile Setup: Users can add their name, location, and a profile picture to personalize their account.

Password Recovery: Users can reset their password if they forget it using a simple email link.

Account Settings: Users can update their contact details or delete their account if needed.

Module 2: Skill Listing
This module helps users to post skills they want to trade.

Create listing: Users can add a title and description of the skill they're offering.

Category Selection: Users can choose a category (e.g., graphic designing, coding, logo making) to make their listing easy to find.

Module 3: Skill Discovery and Matching

Search Skill: user can Browse and search for skills.

Suggested Skills: user can view suggested users based on skill match.

Module 4: Skill Exchange Requests

Send Request: user can send a request to connect for skill exchange.

Accept Request: user can accept or decline incoming requests.

View History: user can view request history and status.

Module 5: Scheduling

Create Sessions: user can propose and accept session timings.

View Sessions: user can view scheduled sessions on a calendar.

Module 6: Feedback

Reviews: user can write and view reviews.

1.5.2 Admin Web Modules

These modules are built for system administrators to monitor and manage the platform.

Module 1: User Management

Manage Accounts: Admin can view, edit or deactivate user accounts.

Monitor Users: Admin can monitor users reported.

Module 2: Dashboard and Analytics

Analysis: Admin can navigate platform usability statistics.

Track Activities: Admin can check metrics such as amount of skill exchanges or active users or top-rated skills.

Module 3: Announcements

Announce Updates: Admin can announce changes or updates in the system.

1.6 Related System Analysis/Literature Review

Several other existing platforms support skill learning or exchanges, but they either focus on paid services or have limited features. Below is a comparison highlighting main weaknesses in related systems and how Barter Brains addresses those gaps

Website name	Weakness	Proposed Project Solution
Skillshare	Requires users to pay for access to courses; no real-time person-to-person exchange.	Barter Brains offers free, mutual skill exchange between users without involving money.
Meetup	Not designed specifically for learning; focuses on group events and has limited skill matching.	Barter Brains focuses on one-on-one skill exchange with intelligent skill matching features
Tandem	Limited to language exchange only; does not support other skills like art	Barter Brains supports a wide variety of skills, not limited to language learning.

1.7 Vision Statement

Barter Brains aims to build a helpful and accessible platform where people from all backgrounds can connect to exchange skills without using money. The goal of Barter Brains is to encourage a learning culture based on sharing, collaboration and mutual growth. For individuals who want to learn and teach skills without paying and who need an easy way to find others with matching interests, Barter Brains is a web skill exchange platform that connects users for one-on-one learning sessions, with features like skill matching, scheduling and reviews. Unlike paid learning apps or unstructured social media groups, our product offers a free, organized and secure system for skill swapping that is built around user needs and mutual benefits.

1.8 System Limitation and Constraints

1.8.1 Limitations

Internet Dependency: The platform requires a stable internet connection to function. If users have slow internet, they may not able to access the services.

Third-party Tool Usage: Barter Brains does not include built-in video calling, users must rely on external tools (like Zoom or Google Meet) for live sessions.

Users Skill verification: The system depends on user honesty and peer ratings, as it cannot independently verify the actual skill levels of users.

User Adoption: People may be hesitant to trust a new bartering platform, especially for sharing skills or services, due to privacy or safety concerns.

1.8.2 Constraints

Project Scope: The platform will focus only on skill bartering (e.g. teaching a language for graphic designing) and will not include physical goods or monetary transactions in it.

Geographic Reach: The first release will target users in specific regions (e.g. Pakistan) to test the platform before expanding globally.

Feature Set: To lunch quickly, the platform will include basic features like user profile, skill listing and reviews, but advanced features like audio/videos calls will come later.

Budget and Time: The project must be developed within a fixed budget and lunched within 6 months, which limits the features and testing we can do initially.

1.9 Tool and Technologies

Tools & Technologies	version	Rationale
Visual Studio Code	1.89+	Lightweight and efficient code editor for writing code.
Html	5	Basic structure of web-pages.
Bootstrap	5	For creating responsive and mobile-friendly UI designs quickly.
Javascript	ES6+	Adds interactivity and dynamic behavior to web pages.
PHP	8.x	Server-side scripting language for handling backend logic.
Laravel	10.x	A modern PHP framework that support MVC architecture, making backend development more organized and efficient.
MySQL	8+	Relational database system to store user and skill-related data.
XAMPP	8.2.x	Provides an easy-to-use Apache server environment for PHP and MySQL during local development.
CSS	3	Styling system used to define the visual presentation of user and skill- related data

		on web interfaces.

1.10 Project Deliverables

Planning Document: A detailed plan that outlines the Barter Brains goals, timeline and resources needed to build Barter Brains. It helps keep the team on track.

SRS (software Requirement Specification): A document that explains what the Barter Brains platform should do, like user features (e.g. creating profile, search skills) and technical needs (e.g. security, speed).

Design: Sketches and layouts of how the Barter Brains website will look, including screens for users profiles, list of skills and searching of skills.

Prototypes: A basic working model of Barter Brains that shows how users interact with the platform like signing up or requesting for a barter.

Project: The final Barter Brains platform, fully built and ready to use, where users can barter skills without money.

Project Report: Summary of entire project, explaining what we did, challenges we faced and how the platform works, including results and lessons learned.

1.11 Project Planning

To successfully complete the Barter Brains website, a well-structured plan is created. Gantt chart outlines the main phases, key milestones and duration of each activity.

Gantt chart:

Milestone	Duration	Milestone Description	Required Resources
Planning	15 days	Defines the Barter Brains scope and objectives.	Project manager, team lead, MS word/Excel
Requirement Gathering	25 days	Collect functional and non-functional requirements.	Analysts, Stakeholder interviews
Designing Process	15 days	Design UI/UX.	UI/UX designers,

			client feedback
Frontend Development	40 days	Build web & mobile frontends.	Frontend developers, HTML, CSS, Bootstrap 5, JS, VS Code
Backend Development	45 days	Develop backend logic and database handling.	Backend developers, PHP, laravel, MySQL, XAMPP
Testing	15 days	Perform functional and integration testing.	Tester, Developer
Deployment	25 days	Deploy the system online, prepare final documentation & handover the project	

1.12 Summary

Barter Brains, a skill exchange platform for the exchange of knowledge without using money, was thus introduced in this chapter. The concern raised was that the world has become increasingly expensive in terms of learning resources, while Barter Brains offers a solution through the promotion of free, peer-to-peer skill sharing. The chapter clearly explained the goals, scope and limits of the platform, along with the main parts for users and administrators.We compared current systems to see where Barter Brains can improve, like scheduling sessions effectively and offering a wide range of skills. We also looked at the tools and technologies used, and shared the project timeline using a Gantt chart. Overall, this chapter serves to clarify how the Barter Brains system will function, the merits of the Barter Brains strategy, and its applied methods for successful implementation.

Chapter 2

Requirements Analysis

2. Requirement Analysis

This chapter explains how the requirements or Barter Brains platform were studied and identified. It breaks down what the system needs to do (functional requirements), how well it should do those things (non-functional requirements), and how should connect with users and other systems (interface requirements).

2.1 User Classes and Characteristics

User classes' on the Barter Brains website refer to the different subsets of people who are likely to use the site, with every group having a particular requirement or other characteristics that will drive how the platform should work for them.

Table 2-1-1

User classes	Users Characteristics
Skill Seekers	These are people looking for help with something, like learning to code or fixing a resume. They're curious, eager to learn, and want an easy way to find someone who can help them.
Skill Providers	These users have talents to share, like graphic design etc. They're confident in their abilities, enjoy helping others, and want to trade their skills for something useful to them. They need a simple way to show what they can do and connect with others.
Casual Browsers	They're just checking out the site, exploring what skills people offer or what they might learn. They are, curious. They might become Skill Seekers or Providers later.
Skill Builders	These are the users who start out as skill seeker focusing on seeking multiple skills but grow over the time. They are the users who are both active learners and teachers, excited to keep growing and sharing.
Administrators	They're manage the platform, approve user profile, monitor exchanges, handle the complaints and ensures a safe & fair environment.

2.2 Requirement Identifying Technique

This section explains the methods used to gather and define the functional requirements of a barter brains. For example, use cases (which include use case diagrams and detailed descriptions) are very useful for applications that involve user interaction. Each use case helps identify related requirements clearly.

2.2.1 Use case Diagram

2.3Functional Requirements

The functional requirement explains what the system is supposed to do. The main features and functions of the website are known as the functional requirements of the system or a website. All of the features and functions must be according to the needs of the users. The **Barter Brains** provides the different functional requirements to help users teach and learn skills by connecting them with others.

Which are:

2.3.1 Sign-up

Field	Detail
Identifier	Fr-1
Title	Sign-up
Requirement	**User Perspective:** A new user can create an account by entering their name, email, and password, then clicking "Sign Up." **System Perspective:** The system checks if the email is unique, create the account, and send a welcome email. If the email is already in use, it displays, "Email already in use try another."
Source	Project team, accounts are needed to track users and their skills.
Purpose	It keeps the site secure and personalizes the barter experience.
Business Rule	Passwords must be at least 8 characters; emails must be valid formats.
Dependencies	None, this is a starting point for other features.

2.3.2 Sign-in

Field	Detail
Identifier	Fr-2
Title	Sign-in
Requirement	**User Perspective**: A returning user enters their email and password, then clicks "Sign In." **System Perspective**: The system verifies the credentials and logs the user into their account. If the details are wrong, it shows "Invalid email or password."
Source	Project team, users need to access their profiles.
Purpose	Allows secure access to personalized features.
Business Rule	Users must already be registered.
Dependencies	Requires an existing account (Fr-1).

2.3.3 Create Profile

Field	Detail
Identifier	Fr-3
Title	Create Profile
Requirement	**User Perspective**: After signing in, users can fill out profile details like bio, skills, location, and preferences. **System Perspective**: Saves the profile and links it to the user's account.
Source	Project team, necessary for matching and interactions.
Purpose	Builds a personal profile for skill sharing.
Business Rule	All required fields must be filled.
Dependencies	Sign-in (Fr-2).
Priority	High, Essential for creating barter opportunities.

2.3.4 Skill Listing

Field	Detail
Identifier	Fr-4
Title	Skill Listing
Requirement	**User Perspective**: A user can list skills they want to offer, by filling a form with skill name, description, and availability. **System Perspective**: Saves and displays the skill on the user's profile.
Source	Project team, skills are the base of the barter system.
Purpose	Makes skills available to others on the platform.
Business Rule	Skill name and description must not be empty.
Dependencies	Create Profile (Fr-3).

2.3.5 Search Skill

Field	Detail
Identifier	Fr-5
Title	Search Skill
Requirement	**User Perspective**: A user types keywords to find skills offered by others. **System Perspective**: Displays matching results based on keywords.
Source	Project team, for discovering other users' offerings.
Purpose	Helps users find the skills they need.
Business Rule	Only approved profiles appear in search.
Dependencies	Sign In (Fr-2).

2.3.6 Send Request

Field	Detail
Identifier	Fr-6
Title	Send Request
Requirement	**User Perspective**: After finding a skill, the user can send a request to connect. **System Perspective**: Send the request to the other user with detail.
Source	Project team, for initiating barter.
Purpose	Enables user to connect over a skill.
Business Rule	Only send request if logged in.
Dependencies	Search skills (Fr-5).

2.3.7 Accept/Reject Request

Field	Detail
Identifier	Fr-7
Title	Accept/ reject Request
Requirement	**User Perspective**: A user can view requests and choose to accept or reject them. **System Perspective**: update request status and notifies the sender
Source	Project team, decisions needed for engagement.
Purpose	Controls skill-sharing agreement.
Business Rule	Only pending request can be acted on.
Dependencies	Send request (Fr-6).

2.3.8 Session Scheduling

Field	Detail
Identifier	Fr-8
Title	Session scheduling
Requirement	**User Perspective**: Once a request is accepted, user can schedule a time for the session. **System Perspective**: saves and displays session time for both users.
Source	Project team, need for organizing interactions.
Purpose	Aligns user availability for skill exchange.
Business Rule	Time and date must be valid.
Dependencies	Accept request (Fr-7).

2.3.9 Feedback

Field	Detail
Identifier	Fr-9
Title	Feedback
Requirement	**User Perspective**: After a session, users can give comments , reviews. **System Perspective**: store feedback and links it to the user profile.
Source	Project team, help improve the platform.
Purpose	Builds trust.
Business Rule	Feedback allow only after session end.
Dependencies	Session scheduling (Fr-8).

2.3.10 E-Certificate Generation

Field	Detail
Identifier	Fr-10
Title	E-Certificate Generation
Requirement	**User Perspective**: After completing a skill session, the user can request a certificate. **System Perspective**: generate a certificate with names, sills, and completion date.
Source	Project team, add value to learning.
Purpose	Acknowledges learning or teaching efforts.
Business Rule	Session must be mark completed.
Dependencies	Session scheduling (Fr-8).

2.3.11 Sign out

Field	Detail
Identifier	Fr-11
Title	Sign out
Requirement	**User Perspective**: A user can click “sign out” to end the session. **System Perspective**: logs the user out and clear session.
Source	Project team, to ensure security.
Purpose	Ends a secure user session.
Business Rule	None.
Dependencies	Sign in(Fr-2).

2.3.12 User profile Approval(admin)

Field	Detail
Identifier	Fr-12
Title	User profile approval
Requirement	**Admin Perspective**: Admin reviews and approves or rejects a new profile. **System Perspective**: notifies users of approval status
Source	Project team, quality control..
Purpose	Ensures quality and trust.
Business Rule	Admin must review every profile.
Dependencies	Create profile (Fr-3).

2.3.13 Manage Accounts (Admin)

Field	Detail
Identifier	Fr-13
Title	Manage accounts
Requirement	**Admin Perspective**: Admin can suspend, delete or update user accounts. **System Perspective**: applies changes and updates status.
Source	Project team, moderation.
Purpose	Maintain platform integrity.
Business Rule	Only admins can access.
Dependencies	Sign in(Fr-2).Admin role.

2.3.14 Monitor and Handle Reports (Admin)

Field	Detail
Identifier	Fr-14
Title	Monitor and Handle Reports (Admin)
Requirement	**Admin Perspective**: Admin can view reported users or sessions and take action. **System Perspective**: Logs actions and notifies users.
Source	Project team, to handle issues.
Purpose	Protect users and manage abuse.
Business Rule	Reports must have a reason.
Dependencies	Report feature (implicit).

2.3.15 Platform Analysis

Field	Detail
Identifier	Fr-15
Title	Platform Analytics (Admin)
Requirement	**Admin Perspective**: Admin views statistics like active users, sessions, feedback, etc. **System Perspective**: Displays data in dashboard format.
Source	Project team, insights for decisions.
Purpose	Track platform performance.
Business Rule	Data must be anonymized.
Dependencies	System logs.

2.3.16 Communication with Users(Admin)

Field	Detail
Identifier	Fr-16
Title	Communication with Users (Admin)
Requirement	**Admin Perspective**: Admin can send messages or announcements to users. **System Perspective**: Delivers message to user email.
Source	Project team, announcements or help.
Purpose	Keep users informed.
Business Rule	Only admins can send mass messages.
Dependencies	Sign-in (Fr-2), Admin role.

2.4 Non-Functional Requirements

The Non-Functional requirements focus on the quality of the website, focusing on how reliable, easy to use, fast, and secure it should be, rather than specific features (which are covered in functional requirements). The **Barter Brains** provides the different non-functional requirements Which are:

2.4.1 Reliability

The reliability website is dependable. It should not crack often, and when it does, it should be fix quickly. This is important because if the site is down, users can not search for or offer skills, which stop the whole bartering process.

Consequences of Failure: If the site happens to go down, then the users cannot search for any skills which stops trades from happening.

Error Reporting: This crash and bug will be logged by the system and notified to the support team immediately.

2.4.2 Usability

The site should be easy to learn, use, and recover from mistakes, so users feel comfortable trading skills. It is about making website user-friendly, even for the beginners.

Ease of Use: Buttons like “Search” or “Contacts us” will be big, clear, and in the same place on every page.
Error Avoidance: If a user forgets to fill in a field (like a password), the site should highlight it.

2.4.3 Performance

The systems should be quick. When users for a skill, they should see the result within few seconds even if 1000 people are using the same site at the same time.

2.4.4 Security

The website must keep user’s information (like emails, passwords) safe. It means protecting site from and its users from harm. This build trust and users are relaxed while using it that their information is saved.

2.5External Interface Requirements

This section explains how the Barter Brains platform interacts with users, software, hardware, and communication systems. These interfaces ensure that the platform functions smoothly across different environment.

2.5.1 User Interface Requirements

The design of Barter Brains will focus on making things simple, clear, and consistent for users. We’ll follow common design rules to keep everything familiar and easy to use. Fonts will be clean and readable, and all pages will use a similar layout so user don’t get lost.
Every screen will include helpful icons, buttons, and a top menu for navigation. The website will be fully responsive, meaning it’ll work well on desktops, tablets and smartphones.

2.5.2 Software Interface

Barter Brains is built using modern technologies:

- The frontend is built using HTML, Bootstrap 5 and JS.
- The backend is built using PHP laravel.
- All data, such as user profiles and skill exchange, will be stored in MySQL database.

It’s designed to work with all modern web browsers like Google Chrome, Mozilla Firefox and Microsoft Edge.
We add support for video calling using services like Zoom or Google Meet for live skill exchange sessions.

2.5.3 Hardware Interfaces

Since Barter Brains is a website, it doesn't need any special hardware to work. Users will interact with Barter Brains using regular devices such as Desktop, laptop, smartphones and tablets etc.

2.5.4 Communication Interfaces

To keep users informed and connected, the platform will support:

- **Email notifications** for things like skill exchange requests, updates, or alerts.
- **In-app messaging** so users can safely communicate within the platform.

All data exchanged between users and the server will be secured using **HTTPS**. Emails will be handled using standard email protocols (like SMTP), and the platform's front and back end will communicate using **REST APIs.** It's designed to work well over normal internet connections, whether it's Wi-Fi or mobile data.

2.6 Summary

To sum it up, Barter Brains is constructed as a smooth, safe, and simple-to-use platform that is well designed for those who want to exchange skills with others in their community. Important functional requirements such as profile creation, skill searching, and exchange request handling are also outlined.

For planning the interactions with the users, a use case modeling was done, and also considering non-functional qualities such as usability, performance, reliability, and security enhance the performance of Barter Brains. Finally, we detailed the interface requirements such that it explains Barter Brains provides a consistent experience while working seamlessly with different tools, devices, and communication systems. All of these come together in forming a solid and well thought out foundation, which is ready to be turned into actual features and code in the development phase.

Chapter 3

Design and Architecture

3. System Design

Barter Brains is a platform where people can trade various skills with one another, like trading their guitar lessons for cooking tips. It is a web-based platform that can be accessed on a mobile, tablet, or computer without needing anything to be installed. The whole application rests on three key pillars: backend server performing all of the business logic and database operations, a secured database for transactional user skill and exchange records, and a clean, responsive frontend, which adapts to various screen sizes.

3.1 Design Consideration

Design Considerations for the Barter Brains website are the key factors we kept in mind while planning how the platform should work. They include the tools and services it depends on, any limits it might face, and the possible risks such as data security or system overload that we've tried to prepare for in advance.

Assumptions and Dependencies

There are some assumptions on which the design of Barter Brains rests. Primarily, users are expected to have a stable Internet connection with access to desktop and modern browsers, such as Chrome, Firefox, or Edge. It is presumed that users know how to interact online- create accounts, send skill exchange request, and browse profiles. This system depends on third-party services for functioning properly regarding email APIs supporting notifications and external login options available via Google.

Limitations

While it promises to be user friendly and flexible, Barter Brains brings with it some limitations. Outdated browsers or very slow internet connections would not perform well on this platform. At the time of launch, the platform might have only a few basic core skill categories and limited language options; therefore, accessibility issues could arise for some users. Also, since it depends on third-party services (e.g., for login and notifications), any issues or downtimes with those services could affect the user experience.

Risks

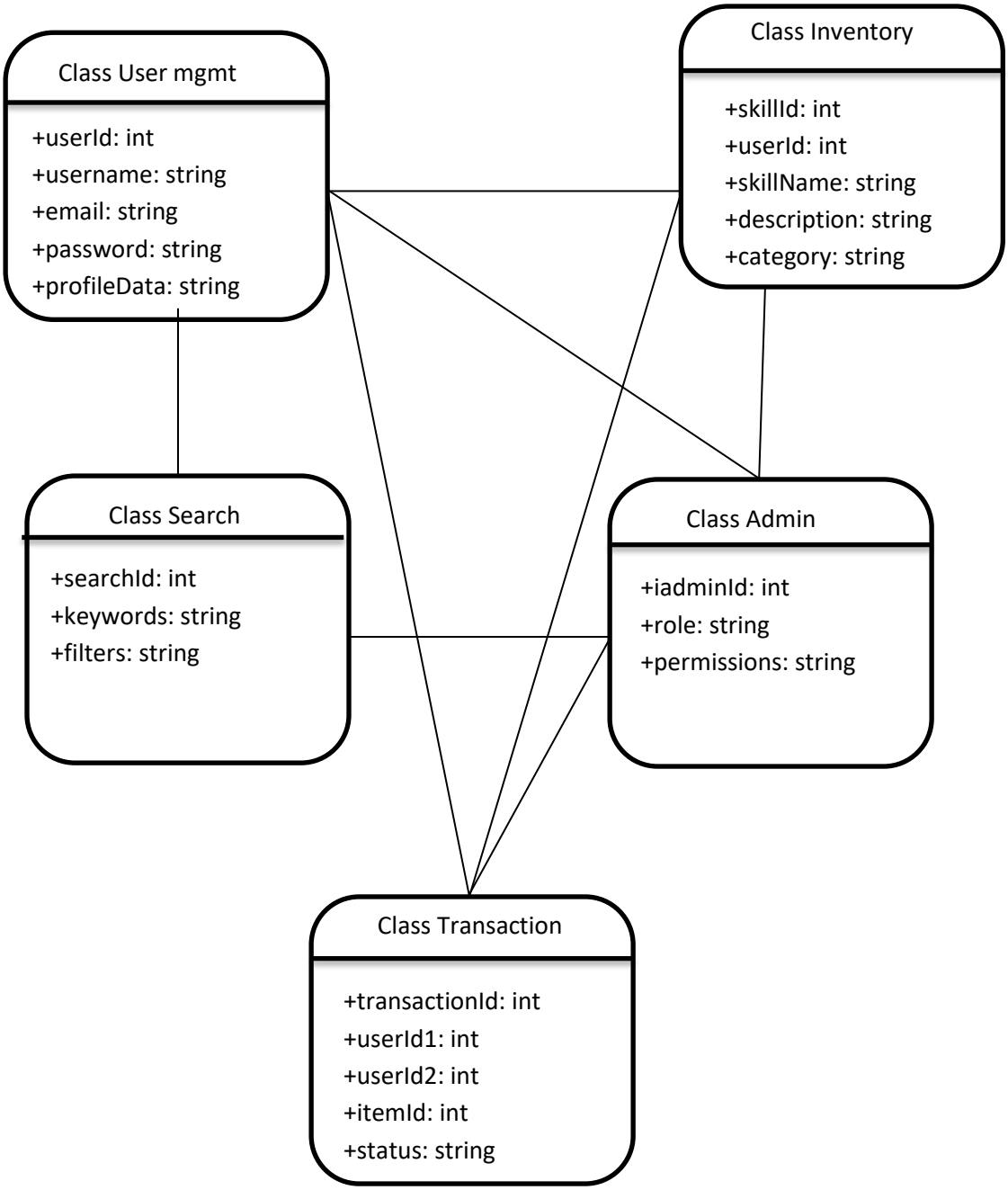
Several potential risks were considered during the design of Barter Brains. The first is that of data privacy since users are now feeding in personal information, the system is designed to be securely authenticated and to store such data encrypted. The second is that of overload of the system, particularly if a large number of users join at the same time. For this aspect, the overall building of the platform comprises the idea that such a platform will grow without speed decay.

3.2 Design Models

To better understand how Barter Brains is structured and functions, we've used several object-oriented models. These help visualize the different parts of the system, how they interact, and how they behave over time. Below are the descriptions of the models used:

3.2.1 Class Diagram

The class diagram shows the building blocks of Barter Brains in the term of objects, their attributes, and how they relate to each other.



Explanation of Class Diagram

This diagram shows different parts (called classes) of a system, likely for a trading or bartering platform, and how they connect. Here's a simple explanation:

Class User Management: This part handles user details like a user ID, username, email, password, and profile information.

Class Inventory: This keeps track of items for trading, with details like item ID, the user who owns it, item name, description, and category (e.g., electronics or clothing).

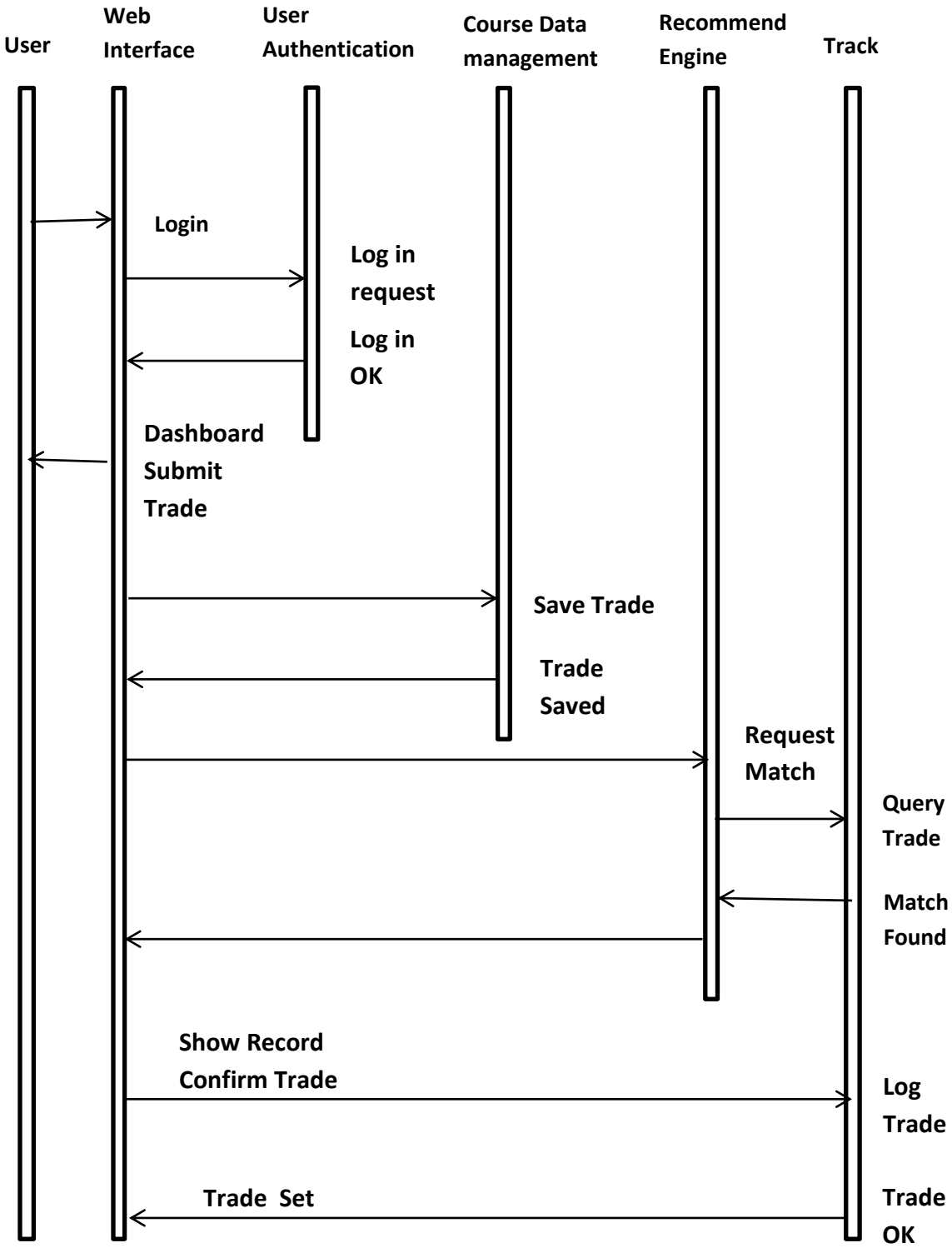
Class Search: This helps users find items or people by using search keywords and filters (like price or category).

Class Admin: This manages admin users, with details like admin ID, their role (e.g., moderator), and permissions (what they're allowed to do).

Class Transaction: This tracks trades, including a transaction ID, the two users involved (user1 and user2), the item being traded, and the status (e.g., completed or pending)

3.2.2 Interaction Diagram (Sequence Diagram)

We used a **sequence diagram** to show how different parts of the system communicate during a typical user action like sending a skill exchange request



Explanation of Interaction Diagram:

- **User Logs In:**

The User accesses the site through the Web Interface.

The system sends a login request to User Authentication.

If the credentials are correct, the system replies with Login OK.

- **User Submits a Skill Trade**

After login, the user accesses their dashboard and submits a trade (offering a skill and requesting one).

This trade data is sent to Course Data Management to be saved.

Once saved, a confirmation (Trade Saved) is sent back.

- **Skill Match is Requested**

The system now requests a match for this trade from the Recommendation Engine.

The Recommendation Engine checks the database (Track) for compatible trades.

If a match is found, it sends back a recommendation (Rec: Bop).

- **User Confirms Trade**

The system shows the recommended trade to the user.

The user confirms the trade.

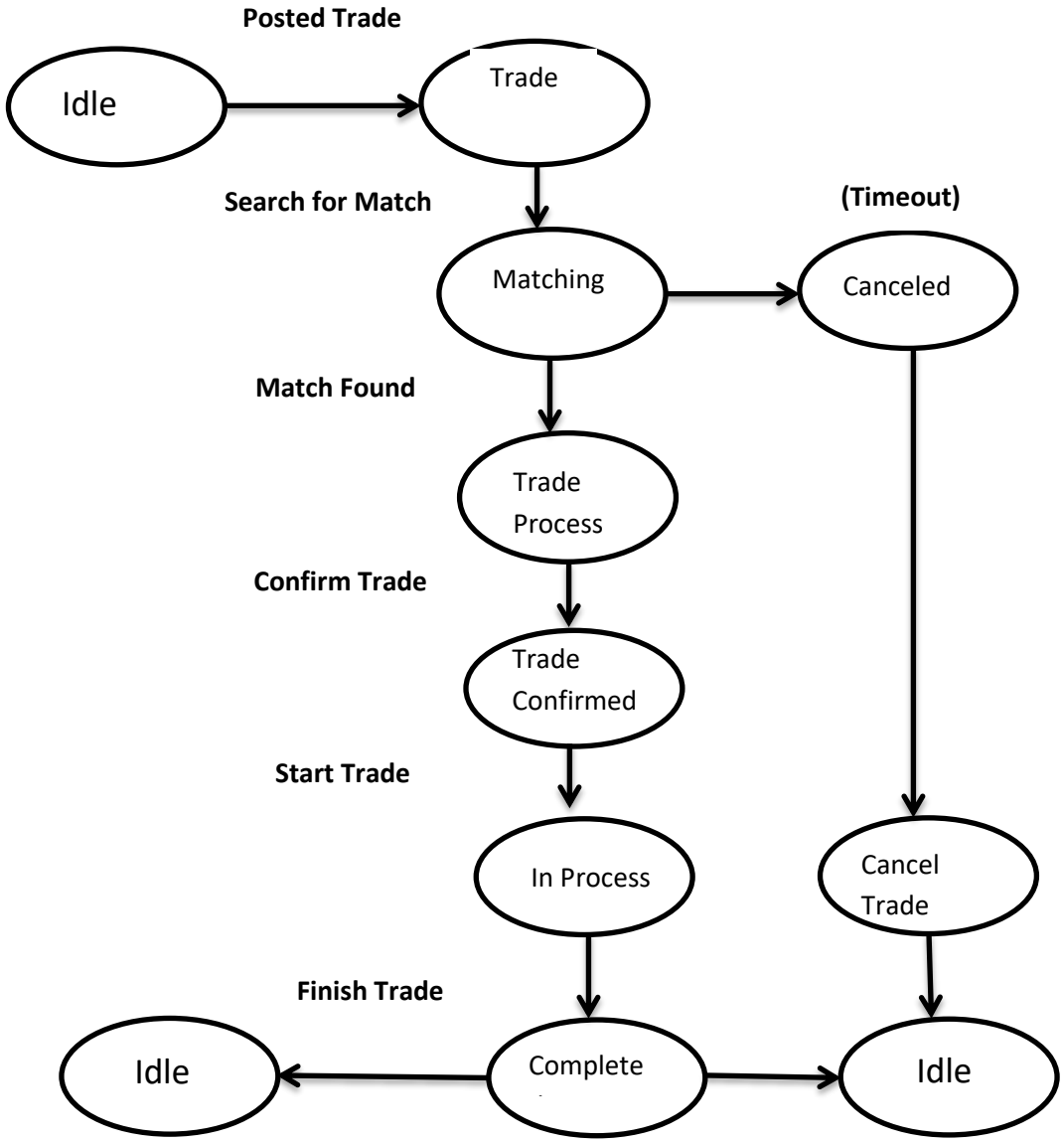
- **Trade is Finalized**

The confirmed trade is then logged in the system.

The system sends a Trade OK response, confirming that the trade is now active/set.

3.2.3 State Diagram

We also created a state transition diagram for the **ExchangeRequest** object. This diagram helps track how the system reacts to different events and ensures that the exchange flow is managed properly in the backend.



Explanation of the State Diagram:

Idle: The user is doing nothing yet.

Trade Posted: The user posts a skill they want to offer.

Matching: The system searches for someone whose needs match the posted skill.

Canceled (if timeout): If no match is found in time, the trade is canceled.

Trade Process: A match is found! Now the users discuss or prepare for the trade.

Trade Confirmed: Both users agree to do the skill exchange.

In Process: The actual skill exchange (teaching, tutoring, etc.) is happening.

Completed: The trade is finished successfully.

Back to Idle: The user can now start a new trade if they want.

Cancel Trade: If something goes wrong or someone backs out, the trade can be canceled and the user returns to idle.

3.3 Architecture Design

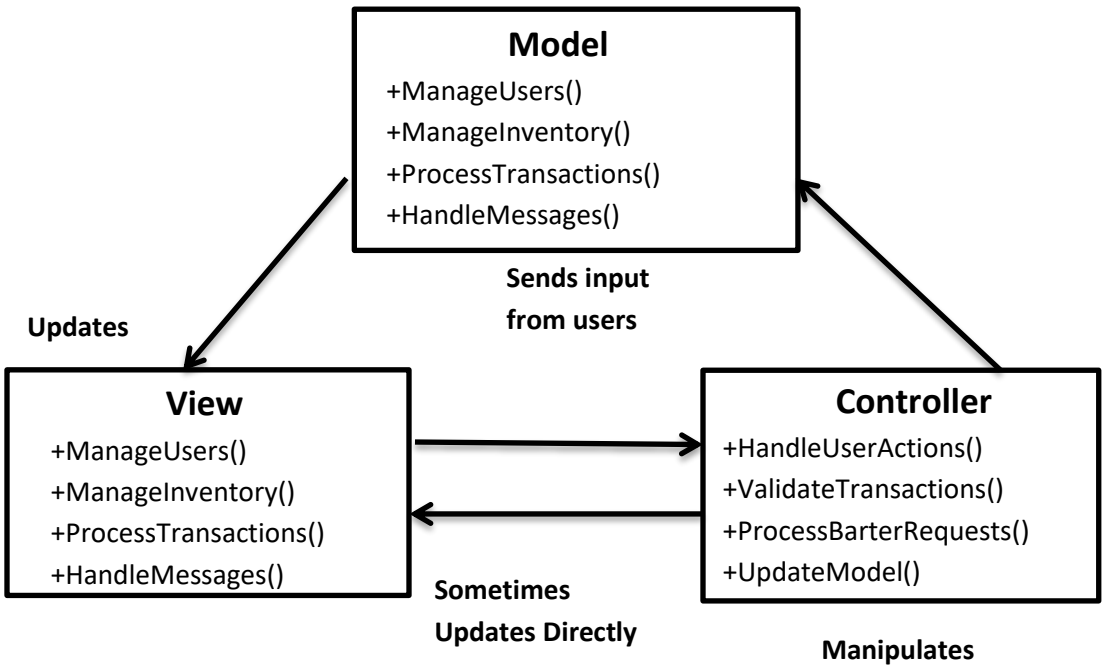
Barter Brains architectural design is a multi-layer architecture based on the MVC or model-view-controller design pattern. Thus, it segregates concerns - user interface, data, and logic into independent blocks-and is client-server architectures in which the client sends requests while the processing of data and returning results are done at the server end.

3.3.1 MVC for Barter Brains

MVC is a software architectural pattern that separates an application into three interconnected components:

Components:

1. **Model:** Represents data and business logic of an application.
2. **View:** Represents the user interface and displays the data provided by the model.
3. **Controller:** Receives inputs and acts as an intermediary between model and view.



3.4 Data Design

Barter Brains' data design specify how all essential information storage, organization, and processing of users, skills, messages, and exchange requests happens. All data is hosted on a well-defined database management system (like MySQL), and the system follows an object-oriented design where everything revolves around classes (objects) with related data (attributes) and behavior (methods).

Clear objects have been defined for users, skills, exchanges, and communication. Each of these has specific attributes describing it (for instance, a user's name or a skill's category), and contains methods for it to perform certain actions (sending or accepting request capabilities).

3.4.1 Data Dictionary

This dictionary catalogs the core entities and data elements of the Barter Brains system to keep focused clarification to developers and stakeholders. Below is the dictionary, in alphabetical list format, of system entities or major data for Barter Brains website in a two-column format. It contains the data type along with brief purpose description in the system.

Terminology	Data Type & Description
ExchangeRequest	Object: Represents a request to exchange skills between two users.
ExchangeRequest.requestID	Integer: Unique identifier for the request.
ExchangeRequest.senderID	Integer: ID of the user who initiates the request.
ExchangeRequest.receiverID	Integer: ID of the user receiving the request.
ExchangeRequest.skillOffered	

	Skill: The skill the sender is offering.
ExchangeRequest.skillRequested	Skill: The skill the sender wants in return.
ExchangeRequest.status	String: Tracks request status (e.g., "Pending", "Accepted", etc.).
ExchangeRequest.date	DateTime: Date the request was created.
Skill	Object: Represents a skill a user can offer or request.
Skill.skillID	Integer: Unique identifier for each skill.
Skill.title	String: Name of the skill (e.g., "Guitar").
Skill.category	String: Skill type (e.g., Art, Language).
Skill.description	Text: Brief details about the skill.
Skill.ownerID	Integer: ID of the user who owns this skill.
User	Object: Represents a platform user.
User.userID	Integer: Unique identifier for each user.
User.name	String: Full name of the user.
User.email	String: Email address used for login and contact.
User.password	String: Encrypted password.
User.skills	List<Skill: All skills linked to the user.

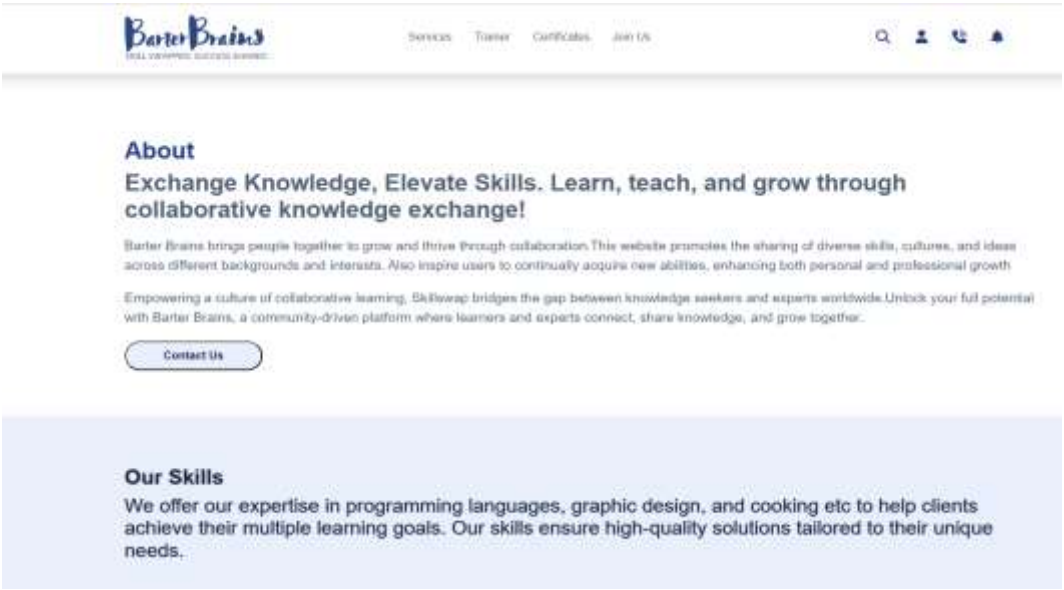
3.5 User interface design

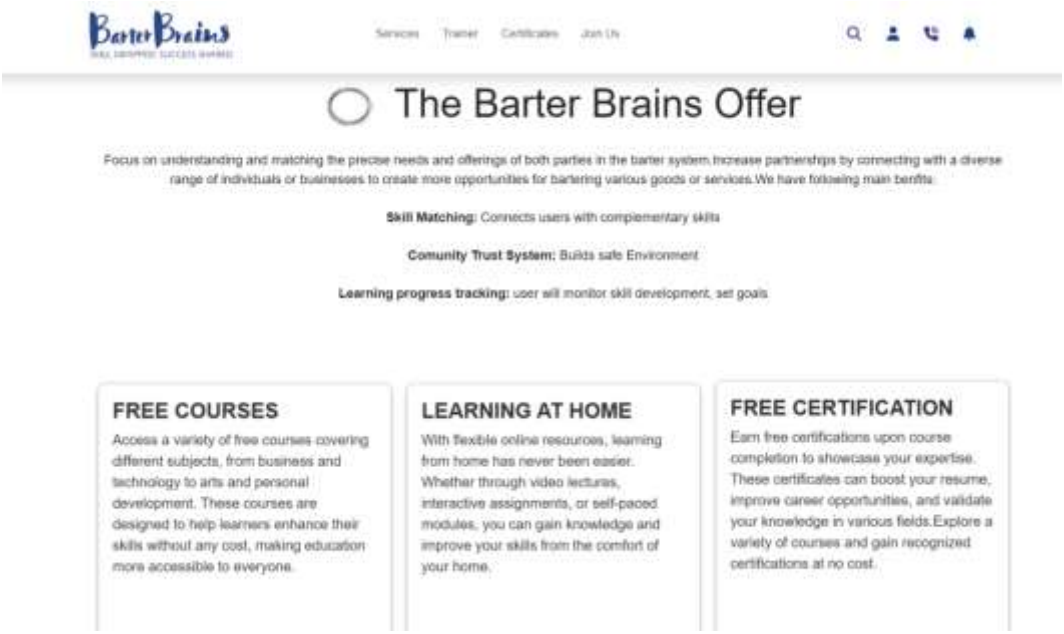
The Barter Brain is user-friendly system, which make it possible for everyone to interact with this system easily. The complete number of web pages are designed and styled using HTML, CSS and Java Script, which are also liked up with each other. The whole interface is responsive according to multiple screen sizes.

3.5.1 User Experience Overview:

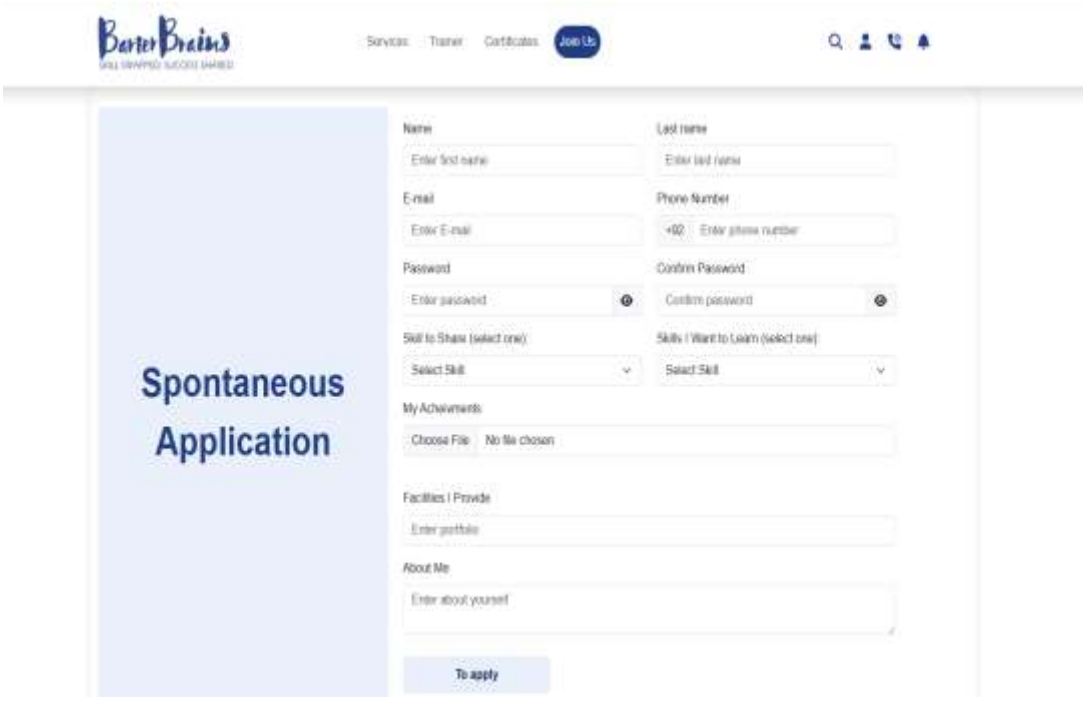
Following are the web pages of Barter Brains:

- **Home (Index) Page:**

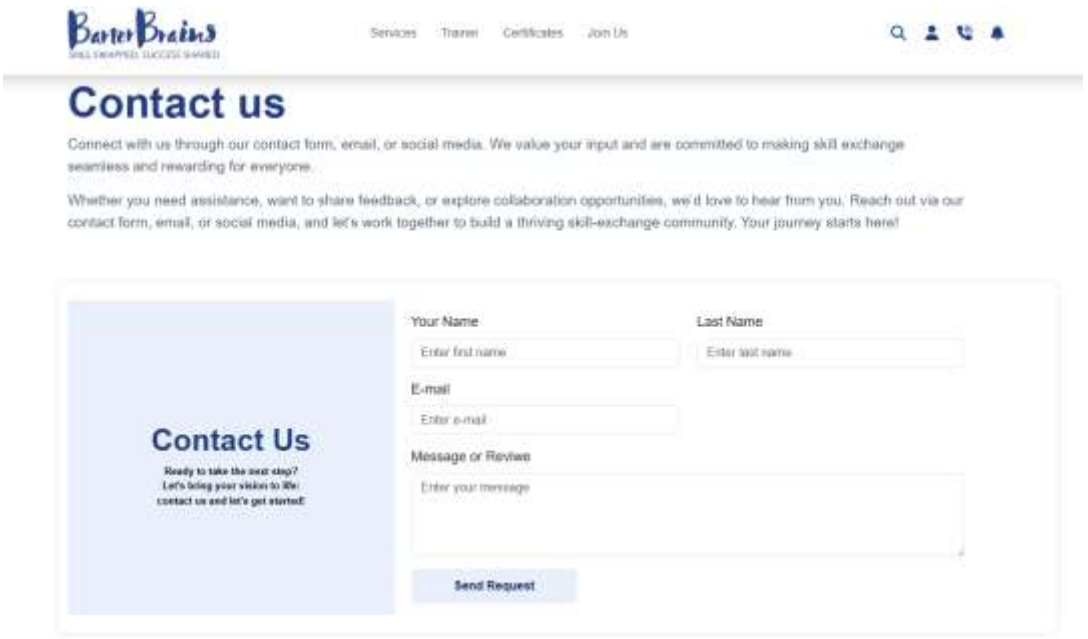




- **Join Us Page (Sign-up):**

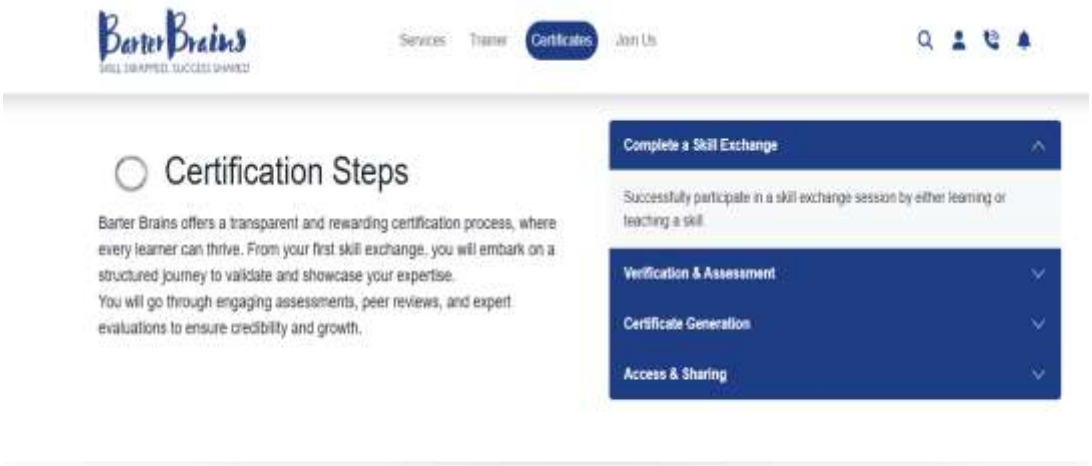


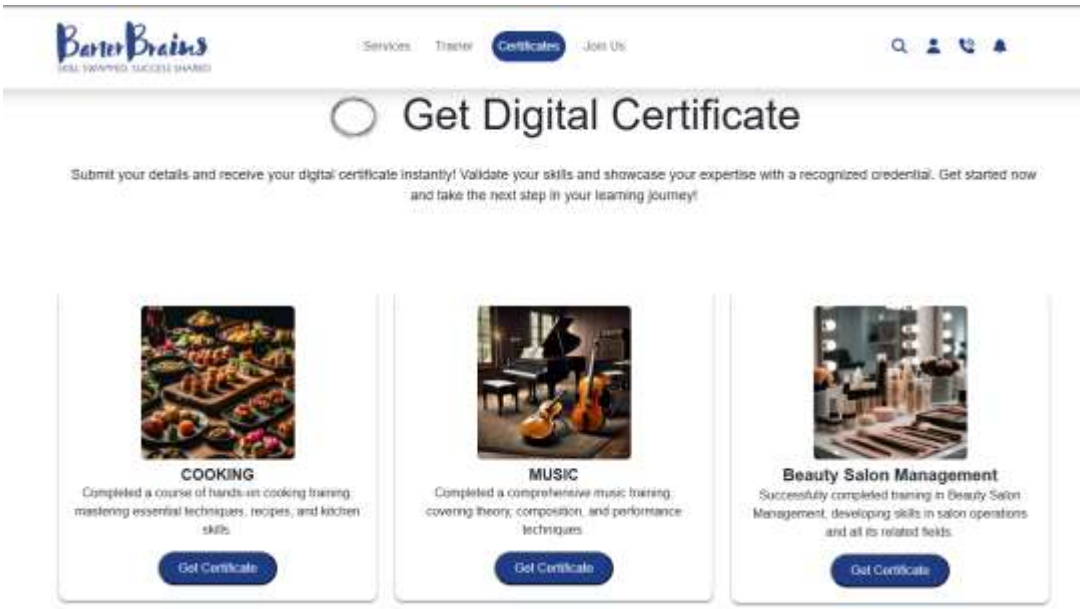
- **Contact Us Page:**



- **Certificate Page:**

  Offer users certificates after completing a session.





- **E- Certificate**

Chapter 4

Implementation

4.1 Algorithms

Barter Brains come with a host of smart features like skill matching, exchange requests, and feedback. some examples below indicate pseudocode for the barter brains website, which help users to find the best possible skill exchange partner based on offering and requiring skills.

Table 4- 1 Skill Matching Algorithm

Input: A list of users with the skills they offer and the skills they need
Output: A list of user pairs that match each other's skills

1: Start with an empty list called `matchedPairs`
2: Go through each user one by one
3: Compare each user with every other user that comes after them
4: If user A offers a skill that user B needs
5: **and** user B offers a skill that user A needs
6: Add this pair of users to the `matchedPairs` list
7: Return the final list of matched pairs

Explanation:

- **Steps 2–3:** Check every possible pair of users
- **Steps 4–5:** Match if both can help each other
- **Step 6:** Save the matched pair
- **Step 7:** Return all matching pairs

Time Complexity: $O(n^2)$ – Because we're comparing every user with others

Table 4- 2 Skill Exchange Algorithm

Input: Pairs of users matched from the Skill Matching Algorithm
Output: A list showing whether their session is confirmed or pending

1: Start with an empty list called `skillExchanges`
2: Go through each matched pair
3: Take the first user from the pair (userA)
4: Take the second user from the pair (userB)
5: If both users are available at the moment
6: Create a skill exchange session between them
7: Add a message to the list showing the session is "Created"
8: Else (if one is busy)
9: Add a message to the list showing "Pending – User Busy"
10: Return the list of all session results

Explanation:

- **Steps 2–4:** Go through each pair and identify the two users
- **Step 5:** Check if both are free
- **Step 6:** Start the session if they are
- **Step 7:** Mark the session as successful
- **Steps 8–9:** If not, mark it as pending
- **Step 10:** Return the full list of session statuses

Time Complexity: O(n) – Only one loop over the matched pairs

Table 4- 3 Feedback Algorithm

Input: Feedback rating from a user after a session
Output: Updated average rating of the user

1: Start a function called `updateRating` that takes in the user and new feedback
2: Get the user's current rating
3: Get how many sessions the user has completed
4: Calculate the new average rating based on the old rating and new feedback
5: Update the user's rating with this new value
6: Increase the number of sessions the user has done by 1
7: Return the updated rating

Explanation:

- **Steps 2–3:** Get the current rating and session count
- **Step 4:** Recalculate the average using the new feedback
- **Steps 5–6:** Save the new rating and update the session count
- **Step 7:** Return the new rating

Time Complexity: O(1) – It uses basic math and does not loop

4.2 External APIs

APIs (Application Programming Interfaces) are tools that enable communication between different software systems. In Barter Brains, APIs are the ones that connect customized features such as user login, notifications, and image upload without having to build these from scratch and thereby make the application smarter and more efficient.

Name of API and version	Description of API	Purpose of usage	List down the API endpoint/function/class in which it is used
Firebase Authentication (v9.0)	A Google-backed service that allows users to sign up and log in securely.	To handle user authentication (sign up, login, logout).	`firebase.auth()`, `signInWithEmailAndPassword()` `createUserWithEmailAndPassword()`
Send Grid API (v3)	Cloud-based service used to send emails easily and reliably.	To send skill exchange confirmations, notifications, and feedback emails.	`sendgrid.send()`, `mail/send` endpoint
Cloudinary API (v1)	Provides cloud storage and optimization for images.	To upload and display profile pictures or certificates.	`cloudinary.uploader.upload()`, `cloudinary.image()`

4.3 Code Repository

To manage versioning and facilitate collaboration among the Barter Brains team, GitHub was used as the primary code repository. All relevant project files such as source code, documentation, and media files were put into that repository. Tracking changes, organizing tasks, and keeping the project current was made easier by using GitHub.

Git Repository Link:

https://github.com/EmanMalik47/Barter-Brains

4.3.1 Metrics of the Git Repository

Commits: A total of **58 commits** were made during the project. This tracked every update in the codebase, showing the consistency and frequency of development work.
Branches: 5 branches were created for major modules such as user authentication, profile management, skill matching, UI layout, and feedback system.
Pull Requests: 12 pull requests were opened for merging feature branches into the main branch. Each PR was reviewed before merging to ensure stable and tested updates.
Issues: 9 issues were recorded (**3 open** and **6 closed**) for tracking bugs, improvements, and development tasks.
Contributors: 3 contributors actively participated in the project, with contributions such as code commits, file uploads, issue handling, and documentation updates.
Code Reviews: 10 code reviews were conducted on pull requests to maintain coding standards, ensure quality, and catch potential bugs before merging.

4.4 Summary

The working smartly of Barter Brains was elaborated into simple yet effective algorithms, strong third-party APIs, and systematic team collaboration through GitHub in this chapter. These elements are the backbone of the platform in making skill exchange faster, secure, and easier for users. Such technical structures not only make Barter Brains' functioning flawless but also achieve the prime objective of portable skill exchange among people.

Chapter 5

Testing and Evaluation

5. Introduction

We perform software testing operations at Barter Brains to ensure the correct operation of the Barter Brains platform. By using this software testing, we are able to check if everything performs as expected and meets the requirements. Automated tests are preferred more often than not; however, there are still cases for manual testing, so all tests should comply with these major principles:

Independent Tests: Each test can run on its own without relying on others.
Clear Assertions: Tests check that the actual results match the expected ones.
Test Data Management: We set up and clean up test data properly.
No Hardcoding: We use variables and configurations instead of fixed values.
Repeatable Tests: Tests can be run multiple times without issues.

5.1 Unit Testing

Barter Brains uses unit testing on members and components which are supposed to function correctly on their own. This catches bugs early on in the development process.

Table 5-1-1

Testcase ID	**UT1**
Requirement ID	REQ-LOGIN
Title	Validate login logic
Description	Test login function in isolation
Objective	Ensure correct handling of credentials
Driver/Precondition	Database mock, test client
Test Steps	1. Call login() with valid creds 2. Observe return value
Input	Email: user@test.com Pass: 123456
Expected Results	Login success token is returned
Actual Result	As expected
Test Status	Pass

Table 5-1-2

Testcase ID	**UT2**
Requirement ID	REQ-SIGNUP
Title	Validate signup
Description	Test signup logic
Objective	Ensure unique email enforcement
Driver/Precondition	No existing user
Test Steps	1. Call register() with existing email 2. Observe error message
Input	Email: user@test.com Pass: abcd1234
Expected Results	"Email already exists" error is returned
Actual Result	As expected

Test Status	Pass

Table 5-1-3

Testcase ID	**UT3**
Requirement ID	REQ-PROFILE-UPDATE
Title	Profile Update Validation
Description	Check updateProfile() function
Objective	Ensure profile data is saved properly
Driver/Precondition	User session mocked
Test Steps	Call updateProfile()
Input	New bio text
Expected Results	Data saved
Actual Result	As expected
Test Status	Pass

5.2 Functional Testing (FT)

Functional testing is the process of ensuring that the functionality of the Barter Brains platform meets the requirements. These features include registration, skill exchanges, and notifications.

Table 5-2-1

Testcase ID	**FT1**
Requirement ID	REQ-LOGIN
Title	User Login Functionality
Description	Full flow of login screen
Objective	Ensure login screen redirects to dashboard
Driver/Precondition	Website deployed, user exists
Test Steps	1. Go to login page 2. Enter creds 3. Click login
Input	Email/password valid
Expected Results	Redirect to dashboard
Actual Result	As expected
Test Status	Pass

Table 5-2=1

Testcase ID	**FT2**
Requirement ID	REQ-SEARCH
Title	Skill Search
Description	Search skill by keyword
Objective	Ensure skill matching and suggestions work
Driver/Precondition	User logged in
Test Steps	1. Enter "cooking" in search 2. Click search
Input	Keyword: cooking

Expected Results	List of users with cooking skill shown
Actual Result	As expected
Test Status	Pass

Table 5-2-3

Testcase ID	**FT3**
Requirement ID	REQ-SKILLREQ
Title	Send Skill Request
Description	Send Skill Request
Objective	Ensure request is submitted & notification sent
Driver/Precondition	Users must be registered
Test Steps	1. Open another user profile 2. Click “Request Skill”
Input	Target skill: Guitar
Expected Results	Success message: "Request sent successfully"
Actual Result	As expected
Test Status	Pass

Table 5-2-4

Testcase ID	**FT4**
Requirement ID	REQ-PROFILE-UPDATE
Title	User Profile Update
Description	User updates profile info (bio, etc.)
Objective	Ensure profile updates are saved
Driver/Precondition	User logged in
Test Steps	1. Go to "My Profile" 2. Change Bio 3. Save
Input	New bio text
Expected Results	Success message, updated profile data
Actual Result	As expected
Test Status	Pass

Table 5-2-5

Testcase ID	**FT5**
Requirement ID	REQ-SESSION
Title	Schedule a Skill Session
Description	User schedules a session with another
Objective	Ensure session scheduling and confirmation
Driver/Precondition	Two users, calendar enabled
Test Steps	1. Select date/time 2. Confirm session
Input	Date: May 20, 6PM
Expected Results	Session confirmed, both notified
Actual Result	As expected

Test Status	Pass

Table 5-2-6

Testcase ID	**FT6**
Requirement ID	REQ-FEEDBACK
Title	Feedback Submission
Description	Review of another user after session
Objective	Ensure feedbacks are stored and displayed
Driver/Precondition	Completed session exists
Test Steps	1. User review skill exchange 2. Save rating
Input	Review message
Expected Results	Review saved, shown on profile
Actual Result	As expected
Test Status	Pass

Table 5-2-7

Testcase ID	**FT7**
Requirement ID	REQ-ADMIN-APPROVE
Title	Skill Approval by Admin
Description	Admin approves submitted skill
Objective	Ensure skill gets approved and is visible to users
Driver/Precondition	Admin logged in
Test Steps	1.Admin dashboard 2.Pending Skills 3.Approve
Input	Skill: "Cooking"
Expected Results	Skill approved, appears on search
Actual Result	As expected
Test Status	Pass

Table 5-2-8

Testcase ID	**FT8**
Requirement ID	REQ-ADMIN-BLOCK
Title	Block/Unblock User
Description	Admin blocks a user with inappropriate behavior
Objective	Ensure user is blocked from login
Driver/Precondition	Admin logged in, user exists
Test Steps	1.Admin 2. Users 3. Block user
Input	Email: abc@test.com
Expected Results	User status: Blocked
Actual Result	As expected
Test Status	Pass

Table 5-2-9

Testcase ID	**FT9**
Requirement ID	REQ-ADMIN-REPORTS
Title	View Reports/Feedback
Description	Admin views platform usage reports
Objective	Verify admin can see platform analytics
Driver/Precondition	Admin logged in
Test Steps	1.Admin 2.Reports Section
Input	N/A
Expected Results	Usage stats displayed
Actual Result	As expected
Test Status	Pass

Table 5-2-10

Testcase ID	**FT10**
Requirement ID	Usage stats displayed
Title	Manage Skill Categories
Description	Admin edits skill categories
Objective	Ensure new categories are saved and shown
Driver/Precondition	Admin logged in
Test Steps	1.Admin 2.Categories 3.Add/Edit/Delete
Input	Category: "Graphic Desiging"
Expected Results	Category updated successfully
Actual Result	As expected
Test Status	Pass

5.3 Integration Testing (IT)

Integration testing verifies how the components of Barter Brains interact with one another. For example, does user login work well with skill requests?

Table 5-3-1

Testcase ID	**IT1**
Requirement ID	REQ-SIGNUP/LOGIN
Title	Sign up + Login
Description	Full flow from sign-up to login
Objective	Validate that new users can immediately login
Driver/Precondition	No user exists with email
Test Steps	1. Register new user 2. Log in using same credentials
Input	Email: test@bb.com Pass: qwerty123

Expected Results	Successful login redirect to dashboard
Actual Result	As expected
Test Status	Pass

Table 5-3-2

Testcase ID	**IT2**
Requirement ID	REQ-LOGIN/SEARCH
Title	Login + Skill Search
Description	Test search functionality after login
Objective	Ensure search is only accessible post-login
Driver/Precondition	Valid user credentials
Test Steps	1.Login 2.Go to search 3.Search skill
Input	Skill: Cooking
Expected Results	Skill results shown
Actual Result	As expected
Test Status	Pass

Table 5-3-3

Testcase ID	**IT3**
Requirement ID	REQ-SKILLREQ/NOTIF
Title	Skill request + Notification
Description	Verify request triggers notification
Objective	Ensure skill request notifies receiver properly
Driver/Precondition	Two users created
Test Steps	1. User A sends skill request to User B 2. User B checks notifications
Input	User A to User B, Skill: Guitar
Expected Results	Notification: "New request from User A"
Actual Result	As expected
Test Status	Pass

Table 5-3-4

Testcase ID	**IT4**
Requirement ID	REQ-SESSION/FEEDBACK
Title	Session + Feedback Flow
Description	Session ends then User submits feedbacks
Objective	Ensure session triggers feedback feature
Driver/Precondition	Session completed
Test Steps	Submit feedback after session
Input	Feedback message
Expected Results	feedback stored, shown on profile
Actual Result	As expected
Test Status	Pass

5.4 Performance Testing (PT)

Performance testing is a way to see how the Barter Brains platform withstands certain stressors and measures the response time and the whole system's stability in the presence of many active users.

Table 5-4-1

Testcase ID	**PT1**
Requirement ID	REQ-LOGIN
Title	Login Load Test
Description	Load test on login endpoint using JMeter
Objective	Ensure login API responds under stress
Driver/Precondition	Server deployed
Test Steps	1. Set up JMeter test plan 2. 50 threads 3. Monitor
Input	Valid credentials
Expected Results	Average response time < 1s
Actual Result	Avg: 790ms
Test Status	Pass

Table 5-4-2

Testcase ID	**PT2**
Requirement ID	REQ-SEARCH
Title	Skill Search Load Test
Description	Load test on search endpoint
Objective	Ensure search doesn't slow under load
Driver/Precondition	Search API live
Test Steps	1. JMeter script for "search?query=cooking" 2. Run
Input	Query: cooking
Expected Results	95% of responses < 2 seconds
Actual Result	98% < 1.5s
Test Status	Pass

Table 5-4-3

Testcase ID	**PT3**
Requirement ID	REQ-SKILLREQEST
Title	Skill Request Load Test
Description	100 concurrent requests to skill endpoint
Objective	Ensure system handles bulk requests
Driver/Precondition	Server deployed
Test Steps	1. JMeter plan with 100 virtual users 2. Monitor logs
Input	Skill: "graphic designing"
Expected Results	95% success ratio, no timeout

Actual Result	98% success
Test Status	Pass

5.5 Summary

Different kinds of software testing were carried out to ensure Barter Brains platform functioning and its adherence to requirements. Unit testing was performed to check the functions of the individual components such as login and registration in order to catch any bugs as early as possible. Functional testing validated features such as user login and skill search and the sending of skill requests completely. Integration testing guaranteed that the different aspects of the system, in particular login and skill requests, integrate well. Performance testing was carried out using JMeter to determine the concurrent user handling capacity of the system with particular emphasis on login and search activities. All tests were developed following best practices such as being independent, using correct test data, steering clear of hardcoded values, and ensuring repeatability.

Chapter 6

System Conversion

6. Introduction

System conversion is a journey from the old system to the new, fully developed "Barter Brains" platform. It may also include migrating the necessary data, all features' setup, and the entire smooth functioning of the live system. The transition should, however, be as smooth as possible without much disruption to users.

6.1 Conversion Method

For Barter Brains platform, we have adopted Pilot Conversion method.

Pilot Conversion

The platform was first launched for a small group of users, maybe testers, friends, or selected volunteers, so that we can get live feedback, do observing how the system performs, correct issues, and make small improvements before fully opening it to everyone. This method helped minimize risk and surely made the experience better once the platform was fully opened.

6.2 Deployment

The deployment process involves the full setup of the website in order for Barter Brains to run properly on any machine. This is part of a student final year project and not for a real-world client. Hence, we used Docker to deploy this system locally in a containerized environment. Therefore, the application can easily run and be managed on different systems.
A step-by-step guide to deployment on Docker:

Build the website

First, we finalized and built the frontend (using HTML, CSS, Bootstrap5 & JS) and backend (e.g., PHP laravel) parts of the platform.

Create Dockerfile and Docker Compose

We wrote a Dockerfile for both the frontend and backend, and ran both services together, with the database, using docker-compose.yml

Set Up the Database

We went with MySQL as the database. The initial schema was created using migration commands or through scripts.

Backup and Restore Data (Optional for Sample Data)

Sample data being used for the tests was exported from our local database and restored in the Docker container for real-world simulation.

Run Docker Containers

All the services were started and tested locally with the command docker-compose up build.

Check Deployment

We verified whether all the core features worked – user registration, login, skill posting, exchange request etc.

6.2.1 Data conversion

There is no real legacy data that Barter Brains has because it is a new system, but we still went through a simple process of data conversion while testing to fill in the system with sample user profiles and skills.

Create sample data

Few mock user accounts and skill entries were created in an .sql file for MySQL.

Clean & Validate

Reviewed data ensuring fields such as email address, skill category and description have been populated correctly.

Load into the target system

This .sql file was imported into the Docker-based database using the command: docker exec -i [container-name] mysql -u root -p[password] barterbrains_db < sample_data.sql

Verify data

Logged into the app and checked that all sample users, skills and exchanges were visible as well and functional.

6.2.2 Training

The user manual contains simple guides for new users on how to use Barter Brains, focusing on the two main use cases:

Use Case 1: Sign-in and Post Your Skill

1. Open the Barter Brains website.
2. Click on "Sign in" and create your account.
3. After logging in, go to "My Skills" and click "Add Skill."
4. Fill your skill name category description with experience level.
5. Submit it: now other users can see your skill and send you exchange requests.

Use Case 2: Search and Request a Skill

1. Login to your account.
2. Use the **Search bar** or browse skill categories.
3. Click on a skill you're interested in.
4. If you find a good match, click on **"Request Exchange."**
5. You can view all sent/received requests in your dashboard.

6.3 Testing After Deployment

Post deployment of Barter Brains via docker, quite a number of tests were conducted to ascertain that everything was working normally and to confirm a successful conversion of the system into an operational platform.

1. **Running Test Cases:**
   We executed test cases for all the core features like:

   - User Registration & Login
   - Skill Posting
   - Skill Searching & Filtering
   - Skills Exchange Requests
   - Profile Editing
   - Request Accept/Reject Flow

2. **Checked Database Connectivity:**
   Ensured that the database was properly connected and all data (skills, users, requests) was being saved and retrieved correctly.
3. **Verified UI & Navigation:**
   Clicked through all buttons, forms, and menus to check for broken links or errors.
4. **Test on Different Browsers:**
   The site opened fully in Chrome, Firefox, and Edge to ensure it was operational on all platforms.
5. **Sample User Flow Testing:**
   Built a test user, posted a skill, and performed a skill request from another user to simulate a real exchange.

6.4 Challenges

There were some challenges during the process of making the deployment and operational setup of Barter Brains. Here is a list together with the way we solved them:

Challenge How We Solved It:

DB not connecting in a docker container The container was running before the DB was ready to connect. We added a wait-for-it script for delaying the connection.
Data not restored properly Our .sql file has syntax errors. We cleaned data and then tested import step-by-step.
Frontend couldn't talk to backend The URLs of APIs were wrong because of Docker's internal networking. We updated .env files with Docker-friendly hostnames (like backend :8000).
Some UI elements weren't aligned Minor CSS bugs were fixed, having been responsive and undertaken a full UI review.

6.5 Summary

The deployment of the Barter Brains platform was completed with Docker so that we could set up and run the application in a consistent manner. This included building the frontend and backend, configuring the environment variables and restoring a sample database and running the system in Docker Compose. We then ran all the test cases we could think of to verify that the application was working as it was supposed to. Along the way, we did face some challenges like connection errors to the database, misconfiguration of the environment, and a few tiny bugs in the UI but we also resolved these by tweaking settings in Docker, cleaning SQL files.

Chapter 7

Conclusion

7. Introduction

The chapter wraps up the project. It looks back at the objectives and goals set at the beginning and checks how well they have been realized. It also put cot through some recommendations for future improvement.

7.1 Evaluation

Below is the list of key objectives of Barter Brains. This table describes whether a certain objective was met with the final developed version of the website Barter Brains.

Table 7-1

Objectives	**Status**
Allow users to be able to register and create a profile with skill tags	Completed
Enable users to search for skills offered by others	Completed
Let users send and receive skill exchange requests	Completed
Add a review feature for exchanged skills	Completed
Show recommendations based on user interests	Completed
User-friendly platform with a clean UI	Completed
Secure Entry and Basic User Authentication	Completed
Display a dashboard to manage requests and responses	Completed

7.2 Traceability Matrix

This section illustrates which feature of the Barter Brains system links back to which original requirement mentioned in the Software Requirements Specification (SRS). This ensures that nothing was left out and that every requirement was in fact implemented and tested.

The table below contains:

Requirement ID: A unique ID for each requirement.
Requirement Description: The explanation as to what the requirement is about.
Design Specification: The part/component of the system in which the requirement was handled.
Code: The specific file/module in which the functionality has been coded.
Test ID: The ID of the test case verifying that requirement.

Table 7-2

Requirement ID	**Requirement Description**	**Design Specification**	**Test ID**
R1	Allow user sign in and profile setup	Component: UserProfileManager	UT01
R2	Search for users by skills	Component: SkillSearchEngine	UT02

R3	Send/receive skill exchange requests	Component: Exchange Handler	UT03
R4	Add reviews and ratings for users	Component: Feedback System	UT04
R5	Recommend skills or users based on interest	Component: Recommendation Engine	UT05
R6	Ensure secure login and authentication	Component: AuthModule	UT06

7.3 Conclusion

The Barter Brains project was developed to provide a platform for the exchange of skills instead of money. Making skills inter-exchangeable encourages individual community interaction, learning, and collective growth whereby users teach what they know and learn what they require. Throughout this project, we were true to the goals and objectives set out at the beginning. Every single feature, from user registration to skill matching and dashboard management, was implemented as planned.

Below is a table showing the mapping of each initial objective to the actual functionality provided in the final system:

Table 7-3

Objective Defined	**Functionality Provided**
Users can sign up and build a skill profile	Registration form and skill-tagging on profile
Users can search for other users by skills	Skill search with filters
Users can send and accept skill exchange requests	Request handling system
Users can review each other after skill exchange	Feedback module
System gives skill recommendations based on interests	Recommendation engine using interests/skills
Secure login and authentication	Login/signup with password protection and validation

7.4 Future Work

As the functionality of the core system is complete, the following are potential areas for improvement and new ideas for enhancing Barter Brains in the future:

Location-based matching would enable users to find skill partners in proximity.

A mobile-app version would facilitate easy access and convenience.

There should be a messaging or chatting system for easy communication between users.

Recommendations based on machine learning so that user matching can be done more precisely.

To enable users who do not speak English to utilize the platform.