|  |  |  |
| --- | --- | --- |
| **Group Number: 15** | | 15 |
| **Project Title:** | Tutord – Skill sharing platform | |
| **Date:** |  |  |
|  | **Name** | **Email** |
| **Group Members** | Patrick Dunne |  |
|  | Allan Smith |  |
|  | Lewis Hammond |  |
|  | Kieran Richardson |  |
|  | Christie Seery |  |
|  | Jack Moffat |  |
|  | Usman Haider |  |
| **URL of shared Logbook** | <https://caledonianac-my.sharepoint.com/:w:/g/personal/pdunne300_caledonian_ac_uk/EfNI63DJertGqmem0GEOdO8BAllS1Ly4NXsRleJlmOb6nA?e=B3dh5j> | |

**Executive Summary**

This report outlines the development of Tutord, a prototype skill-sharing web application created by 7 Computing Students from Glasgow Caledonian University as part of their 12-week integrated project module. Tutord connects users (Students or Tutors), offering features such as course booking, filtering, and user account management. Key requirements included

A responsive web design, a structured rating system, and front-end and back-end functionality are implemented seamlessly.

The application was developed using HTML, CSS, JavaScript, Bootstrap, Django, and SQL, with the database hosted on Microsoft Azure. While the front-end and back-end could not be fully integrated due to compatibility issues, the project delivered a functional front-end prototype, showing the website's potential. This report details the challenges faced, particularly with integration, and the lessons learned for future development [1] [2] [3] [4] [5].

**Table of Contents**

1 Project Introduction 7

1.1 Integrated Project 3 Concept 7

1.2 Project Overview 7

2 Group Organisation and Roles 8

2.1 Role Purpose 8

2.1.1 Front End Developer 8

2.1.2 Back End Developer 8

2.1.3 Report Writers 8

2.1.4 Testers 8

2.1.5 User Interface Designers 8

2.2 Member Roles 9

2.3 Role Changes 9

3 Design & Development Tools 9

3.1 Project Management Tools 9

3.1.1 GitHub 9

3.1.2 Trello 10

3.1.3 Discord 10

3.1.4 MS Teams 10

3.2 Design Tools Description and Justifications 10

3.2.1 MS Word 10

3.2.2 MS Excel 10

3.2.3 Figma 10

3.2.4 Balsamiq 11

3.3 Development Tools/Environments 11

3.3.1 Visual Studio Code 11

3.3.2 Google Chrome and MS Edge 11

4 HCI Approaches 12

4.1 General Aspects 12

4.1.1 Colour 12

4.1.2 Navigation 12

4.1.3 Accessibility 12

4.2 Page Layouts 12

5 Artefact Creation 13

5.1 Development Technologies 13

5.1.1 HTML 13

5.1.2 CSS 13

5.1.3 JavaScript 14

5.1.4 Bootstrap 14

5.1.5 Django 15

5.1.6 SQL 15

5.1.7 Microsoft Azure 15

5.2 Developing the Application 16

5.2.1 Planning 16

5.2.2 Design 16

5.2.3 Development 16

5.2.4 Testing 17

6 Functional Testing 17

6.1 Unit Testing 17

6.2 Integration Testing 17

6.3 User Acceptance Testing 17

7 Review of Final Deliverable 18

7.1 Navigation 18

7.2 Homepage 18

7.3 Courses Page 18

7.4 Sign-Up and Sign-In Pages 19

8 Conclusions 19

9 Appendix 1 – Product Screenshots 20

9.1 Sign In 20

9.2 Sign Up 20

9.3 Home Page 21

9.4 Courses 21

9.5 Cart 22

9.6 Checkout 22

9.7 Learner Profile Page 23

9.8 Tutor Profile Page 23

9.9 Digital Art Course Page 24

9.10 Meal Prep Course Page 24

10 Appendix 2 – Design Wireframes and Hi-Fidelity Examples 25

10.1 Desktop Wireframes 25

10.1.1 Sign Up Page (Tutor) 25

10.1.2 Sign Up Page (Learner) 25

10.1.3 Tutor profile 26

10.1.4 Courses Page 26

10.1.5 Profile Page (Learner) 27

10.2 Figma Interface Designs 27

10.2.1 Log In Page (Tutor) 27

10.2.2 Log In Page (Learner) 28

10.2.3 Create Account Page 29

10.2.4 courses Page 30

10.2.5 Create Course Page 31

10.2.6 Learner Profile Page 32

10.2.7 Course Information Page 33

10.2.8 Tutor Profile Page 34

11 Appendix 3 Acceptance Tests 35

11.1 Create, Sign-In, and Delete Account. 35

11.2 Courses Pages 36

12 Entity Relationship Diagram 40

13 Appendix 4 UML Diagrams 41

13.1 Use Case Diagram 41

13.2 Activity Diagrams 43

13.2.1 Sign Up 43

13.2.2 Log In 44

13.2.3 Delete Account 45

13.2.4 Searching for Course 46

13.2.5 Viewing Course 48

13.2.6 Purchase Course 49

14 References 50

**Tables of Figures**

Table 1 System Roles 6

# Project Introduction

This report outlines the development lifecycle of *Tutord*, a skill-sharing platform designed as part of the Integrated Project 3 (IP3) module by seven students from Glasgow Caledonian University. Turord’s goal is to be an online platform where learners and tutors can connect and share skills in their desired areas, such as Cooking, Art, Finance, and more. The platform supports key features such as booking and filtering courses, user profile management, and rating courses and tutors.

This report documents the group's planning, design, development and testing stages. It highlights the roles within the team, the tools and technologies used, and the processes followed to achieve the project objectives.

To ensure all targets were met throughout the project, the team created an Initial Project Planning Report Document. This document outlined the deliverables, scope, and goals for the project, providing clear guidance throughout the development process.

## Integrated Project 3 Concept

Integrated Project 3 (IP3) is a module specifically designed to provide students with the opportunity to work in groups and complete a project from start to finish. Each group is made up of students with diverse skill sets, which provides a unique learning experience through knowledge-sharing and teamwork throughout the project lifecycle. This module challenges students to address communication, technical and organisational issues while developing a functional prototype or product within the set timeframe.

At the start of the project, the team created an initial Project Planning Report (PPR) document, which served as a crucial resource. The PPR outlined the key objectives, functional and non-functional requirements, deliverables and constraints. The team used the PPR to track progress, set clear expectations, and ensure the project consistently met the module outcomes.

**Report Link:**

[**https://caledonianac-my.sharepoint.com/:w:/g/personal/pdunne300\_caledonian\_ac\_uk/EQobvh8qGCFBidt4YaQHb7QBjJH5pgovueaXDMrwDQ3RUA?e=NdEDhg**](https://caledonianac-my.sharepoint.com/:w:/g/personal/pdunne300_caledonian_ac_uk/EQobvh8qGCFBidt4YaQHb7QBjJH5pgovueaXDMrwDQ3RUA?e=NdEDhg)

## Project Overview

Tutord is a skill-sharing platform designed to connect learners and tutors, providing an easy-to-use experience for finding, booking, and attending tutoring sessions across a wide range of subjects, including Cooking, Art, Finance, and more.

The platform aims to deliver efficient and accessible skill-sharing through a user-friendly design and strong functionality.

The Key deliverables of the platform include:

* **Course Booking and Filtering:** Users can easily search and filter courses by category, rating, difficulty, and Duration, allowing them to find courses specifically tailored to their needs.
* **Rating and Review System:** Users can rate tutors and courses based on their experience, ensuring only quality courses and tutors are on the platform.
* **User Profile Management:** Learners can manage preferences, tack booked sessions, and save favourite courses, while learners can list their qualifications and experience.
* **Responsive Web Design:** The platform functions efficiently across various devices, including desktops, smartphones, and tablets.

Additional deliverables include:

* **Account Creating and Management:** Users can register as either learners or tutors, with role-specific features and secure access.
* **Privacy and Data Protection:** The platform complies with all relevant GDPR laws and regulations, ensuring user data protection.

# Group Organisation and Roles

## Role Purpose

### Front End Developer

All three of the Front-end developers were responsible for building the visual and interactive components of the site. They made sure the design was consistent throughout the whole site and that the layout was responsive to changing screen sizes.

### Back End Developer

Back-end developers were responsible for building and maintaining the server-side functionality and the database management for the site. They ensured that APIs were implemented correctly, data was processed efficiently and managed the integration between the database and the front-end, supporting the functionality of the user-facing components.

### Report Writers

Report writers were responsible for creating concise, clearly written and well-structured documentation for the project. They ensured that the entire development process, from initial conception to final implementation details, and finally the results were recorded accurately and to a high standard.

### Testers

All five testers were responsible for ensuring the reliability, functionality and usability of the system. They assessed both the functional requirements, such as user interactions and operations, and non-functional requirements, including security, and GDPR compliance. The testers identified bugs and errors in the code early, allowing the developers to address the issues quickly rather than at the end of the development cycle, helping to avoid delays.

### User Interface Designers

User interface (UI) Designers were responsible for creating user-friendly and visually appealing layouts for the website. They created wireframes and prototypes using Balsamiq and Figma to establish a visual starting point for the site's navigational flow, structure, and key elements. Their responsibilities included selecting a consistent colour scheme that adhered to accessibility standards, choosing a user-friendly font, designing a suitable layout for the pages, and ensuring the overall design was engaging for the users. Designers also worked closely with the front-end developers to ensure the designs were implemented effectively and accurately, enhancing the user experience for both tutors and learners.

## Member Roles

|  |  |
| --- | --- |
| **Name** | **Roles** |
| Patrick Dunne | Report Writer, Front-end Developer, User Interface Designer, Tester |
| Jack Moffat | Report Writer, Back-end Developer, User Interface Designer, Tester |
| Lewis Hammond | Front-end Developer, User Interface Designer, Tester |
| Allan Smith | Back-end Developer, User Interface Designer, Tester, Logbook Author |
| Christie Seery | Back-end Developer, User Interface Designer, Tester |
| Usman Haider | Front-end Developer |
| Kieran Richardson | Back-end Developer |

## Role Changes

During the project, several role changes took place based on the project's needs. In week 4, a new team member joined and was assigned to the front-end team. They were quickly brought up to speed with the project’s goals, wireframes, and tools by the other members of the team.

Later in the project, the back-end team members who were not writing the report moved over to assist with the front-end to assist with the remaining tasks. This role change helped to improve productivity within the front-end team and ensured the website looked appealing.

In the final few weeks, two members of the team, one from the front-end and one from back-end focused on writing the report. This ensured both perspectives were represented and maintained a consistent writing style, avoiding the potential issues of involving all seven members in the writing process.

# Design & Development Tools

## Project Management Tools

Effective project management was crucial throughout this project to achieve its objectives within the allocated time and available resources. This section will talk in detail about the tools used to plan, track, and manage the project efficiently.

### GitHub

GitHub is an online code repository and version control tool. During the development, A repository was set up for the group, it was used to store different versions of the project across multiple branches, and record contributions from team members, including the date, time and specific changes made. It was chosen for its reliability and familiarity, as all team members had prior experience using it.

### Trello

Trello is an online project management tool based on the Kanban methodology, which uses boards, cards and lists to organise tasks and track their progress. During the development, it was used to assign tasks, monitor progress and set deadlines. Its visual layout and ease of use allowed the team to track the project’s progress briefly and ensure that tasks were completed on time. It was chosen for its effectiveness, simplicity and familiarity among the team [7].

### Discord

Discord is an online communication platform that offers, voice, text, video chat and small-sized file sharing. During the development, the team set up a dedicated Discord server for the project to act as the primary communication hub, with specific channels set up for different topics, roles, and general queries. This structure ensured efficient communication between all team members and shared any updates immediately, eliminating the need to wait for an in-person meeting. It was chosen for its customisable ability, reliability, and familiarity among the team.

### MS Teams

Microsoft Teams is an online communication platform that offers voice, text, video chat, and file-sharing. For this project, the lecturer set up a Microsoft Teams group, which was originally used for communication. However, the team decided to transition to another platform due to limited familiarity and personal preference. It is still used to provide the lecturer with updates and required documentation.

## Design Tools Description and Justifications

Effective design tools were crucial throughout this project to create and enhance visual elements of the site and develop the documentation. This section focuses on the tools used to build a visually appealing and user-friendly site while providing organised and clear documentation.

### MS Word

Microsoft Word is a word-processing tool that offers collaborative editing features, text formatting and templates. During the project, the team used it to create, format and maintain the documentation, including the initial planning report, minutes of the meetings, team roles, and testing documentation. Ensuring professional and well-structured project documentation. It was chosen for its reliability, accessibility, and familiarity among the team.

### MS Excel

Microsoft Excel is a spreadsheet tool that offers data organisation, visualisation and analysis. During the project, the team used it to analyse sample data and create the initial project timeline. Its ability to produce charts, filter data and organise it in tables, ensured reliable data and efficient management of the initial project timeline. It was chosen for its simplicity, reliability, and familiarity among the team.

### Figma

Figma is a web-based design tool that is used for creating interactive prototypes and mock-ups of websites and apps. During the project, the team used it to create the initial design based on the previously created wireframes, ensuring the visual layout aligned with the initial planned structure. Its collaborative features allowed the team members to adjust and provide feedback in real-time, instead of at an in-person meeting. It was chosen for its ease of use, extensive features and familiarity among the team.

### Balsamiq

Balsamiq is a software and web-based wireframing tool used for designing app interfaces and websites. During the project, the team used it to create the initial wireframes, focusing on the layout and structure rather than visual details. This helped build a foundation for the final design and eliminate possible options. Its easy-to-use features such as its drag-and-drop allowed the team to create and edit the wireframes collaboratively. This helped to streamline the design process ensuring it matched the project objectives. Balsamiq was selected for its simplicity, effectiveness in visualising early concepts, and familiarity among the team.

## Development Tools/Environments

### Visual Studio Code

Visual Studio Code is a lightweight code editor supporting various programming languages and frameworks. It was the primary development environment for writing, editing and debugging code throughout the project. VS Code’s built-in Git integration allowed for efficient collaboration with regular commits helping track version control effectively. The live share feature further improved teamwork by enabling team members to view and edit code from one another’s screens in real time. VS Code was selected for its user-friendly interface, the vast number of extensions and packages that can be added, and familiarity among all team members.

### Google Chrome and MS Edge

Google Chrome is a popular web browser known for its speed and ease of use. During the project, the team used it to test and debug the website, ensuring the website functionality was correct and consistent design. The front-end team used Chrome’s built-in developer tools to debug JavaScript and inspect different elements, streamlining the development process. Chrome was selected for its easy-to-use features, various development tools, familiarity among the team, and its popularity with possible end users.

Microsoft Edge is a popular web browser with excellent accessibility features such as Read Aloud. During the project, the front-end team used it to verify the website’s appearance and functionality across browsers. It was a reliable tool for cross-browser testing. Edge was selected for its ease of use and availability on all front-end team members’ devices.

# HCI Approaches

## General Aspects

### Colour

A black and white text

Description automatically generated

The colour palette was designed to reflect the simple style of platforms like Skillshare. A combination of light blue (#E0F7FA) and dark grey (#374151) backgrounds creates a visually appealing colour scheme. The light blue provides a cool ocean-like theme, while the dark grey reduces eye strain and provides a strong contrast. White text on these backgrounds meets accessibility standards (W3C, 2022). Colours, such as dark blue (#3ABFF8) and orange (#F97316), were selected to highlight key elements, ensuring an easy-to-navigate and accessible design for all users [2].

### Navigation

To ensure consistent and easy navigation, we implemented a fixed navigation bar at the top of the screen, visible on all pages. This bar allows users to access any part of the site by clicking on its links or the logo, which takes them back to the homepage. On screens smaller than 1200px, the navigation bar collapses into a stacked layout for better usability, with a hamburger menu providing access to a dropdown containing links to all pages. Additionally, a smaller navigation bar is placed at the bottom of the screen with quick links to four key pages, enhancing accessibility on mobile devices. The top navigation bar highlights the current page in #105BD8 (Figure 1), helping users stay oriented and reducing accidental re-navigation. This strategy ensures the navigation remains simple, intuitive, and accessible across platforms. [1] [3] [2].

### Accessibility

Ensuring accessibility in web design is a crucial step in providing an inclusive experience for all users, regardless of their abilities or devices. The website was designed with a responsive layout to adapt seamlessly to various screen sizes, accommodating the growing use of mobile devices. High-contrast colours, as outlined in section 4.1.1, were chosen to enhance readability, particularly for users with visual impairments. By using the Bootstrap framework, we ensured the design and functionality worked smoothly across all platforms, creating an enjoyable and accessible experience for everyone. [2].

## Page Layouts

1. **Sign-Up Page (Tutor and Learner)**

These pages were designed to simplify the account creation process. Both page layouts use a form centred on the page, with clearly labelled fields to minimise confusion and improve accessibility for users. A dropdown menu allows users to select their account type (Tutor or Learner). The repeated structure of the fields ensures that users can accurately enter their details. The simple design makes it easy for users to complete the sign-up process, regardless of the device they’re using.

1. **Login Page**

The login page features a simple design, showing only essential input fields (email and password) located in the centre of the page. A clear “Create Account” link is included for new users, ensuring a seamless user experience. The consistent use of spacing and alignment ensures that the page is functional on all screen sizes and easy to navigate.

1. **Profile page (Tutor and Learner)**

The profile pages and specifically designed to display user-specific information. For tutors, the page includes their name, qualifications, a description of their services, and to select a tier of service. For learners, the page highlights their enrolled courses and account details, with an option to delete their account. These layouts prioritise clear easy-to-access information. Navigation icons for the homepage and account settings are placed at the top of the page for quick access.

1. **Courses Page**

The courses page features a grid layout showing different courses on cards, displaying the course name, tutor, and price. The search bar allows users to filter and find specific skills or courses. The design focuses on a clear organisation of the course information, making browsing straightforward and user-friendly.

# Artefact Creation

## Development Technologies

### HTML

HTML(HyperText Markup Language) is a markup language used to create and structure pages when developing a website. It defines elements including links, paragraphs, images, headings and other media, allowing them to be displayed in a browser. [3].

The purpose of HTML is to provide a structure for organising and displaying content on a webpage. It allows the content to be arranged logically and accessibly, building the overall layout for the website. HTML allows elements to be defined by developers including links, paragraphs, images, headings and other media, while also providing the ability to add forms and interactive media. This allows developers to create interactive user-friendly web pages that are responsive across various browsers and devices.

HTML was selected as it is supported by all browsers and provides a reliable and widely recognised standard for displaying and structuring content on web pages. Its simple syntax and compatibility make it the clear favourite for web development, supported by the team’s prior experience using it.

### CSS

CSS (Cascading Style Sheets) is a stylesheet language used to control the layout and presentation of elements on a webpage. It allows developers to define styles including fonts, colours, spacing and alignment, as well as complicated design features like animations and responsiveness. [3].

The of CCS is to separate the structure of the content written in HTML from its presentation. This allows for a flexible and consistent design across the website, ensuring an appealing design while being easy to maintain. CSS allows developers to style individual elements or apply styles globally using IDs, classes, and selectors. CSS provides responsive design capabilities allowing webpages to adapt to different screen sizes and devices.

CSS was selected because it is supported by all browsers and offers excellent control over the visual aspects of a webpage. Its responsiveness features and compatibility with HTML made it an ideal choice for designing visually consistent and user-friendly websites. The flexibility of CSS allows for the creation of professional designs while maintaining accessibility. The team also selected CSS due to all members having prior experience using it, which helped speed up the design process.

### JavaScript

JavaScript is a high-level object-oriented programming language primarily used to create interactive content on web pages. It allows developers to implement features including interactive animations, form validation, and user interface enhancements. [1].

The purpose of JavaScript is to add functionality and interactivity to webpages, making them more responsive and engaging. JavaScript handles user input, and updates content dynamically without the need for reloading the webpage, allowing form validation and interactions based on events like button clicks. It is essential for creating interactive elements like pop-up windows, dropdown menus, and responsive animations.

JavaScript was selected because of its compatibility with all major browsers. It integrates seamlessly with HTML and CSS, allowing developers to create fully functional, interactive web pages. Its large volume of libraries and frameworks, including jQuery and React further improve its uses. Additionally, JavaScript is widely supported by the web development community, providing almost unlimited resources through discussion forums and meet-up groups. The team also selected JavaScript due to all members having prior experience using it, which helped speed up the design process.

### Bootstrap

Bootstrap is a popular front-end framework used to create responsive grid-based websites. It provides pre-built components, including buttons, navigation bars, windows, and grids, along with CSS and JavaScript integrations to simplify the development process. [8].

The purpose of Bootstrap is to allow for fast development of responsive and visually appealing web pages. Its grid layout allows developers to create webpages that adapt to various screen sizes and devices without interruptions to the user.

Bootstrap was chosen for its ease of use and its ability to create responsive webpages quickly. Its large library of components made it perfect for creating a professional-looking website without the need to build every element from scratch. Bootstrap is compatible with all modern browsers and integrates with HTML, CSS, and JavaScript making a clear favourite for the team. Additionally, several members of the team wanted to learn to use Bootstrap, so this Integrated Project module presented an excellent opportunity.

### Django

Django is a high-level framework used for designing and building websites using Python. It simplifies development by providing various libraries and tools that address tasks like database management and user authentication. Django handles the back-end logic of the website making it an excellent choice for developing large systems.

The purpose of Django is to provide an efficient and reliable framework for managing the back end of a website. It allows developers to build, scale, and maintain websites by simplifying tasks like form validation, database interaction, and user authentication. Django is crucial for websites that require a structured back-end, scalability, and secure data handling.

Django was selected for its popularity, its scalability, and its simplicity. Its built-in tools like the Object-Relational-Mapping system, make it easy for developers to interact with the database and perform complex operations using Python code. Additionally, the back-end team had previous experience with using Python making Django an excellent choice.

### SQL

SQL is a standard programming language used to manage and manipulate relational databases. It allows developers to retrieve, insert, update, and delete data from a database efficiently. SQL also allows for creating and modifying database structures, including tables and indexes.

The purpose of SQL is to provide and reliable way to manage and interact with data stored in a database. It allows developers to carry out complex queries to retrieve specific data, filter and organise information, and ensure the reliability of the stored data. SQL Is essential for applications and websites that rely on databases for storing user information, records, and any other structured data.

SQL was selected for its widespread use, its ability to integrate with various database management systems like MySQL and SQLite, and its simplicity. Its easy-to-read structure syntax makes it simple to perform complex data operations with little code. The scalability and reliability for handling large datasets, along with the familiarity within the team, made it a perfect choice for managing the project's data.

### Microsoft Azure

Microsoft Azure is a cloud computing platform provided by Microsoft. It provides a wide range of services, including virtual machines, databases, storage, web hosting, and artificial intelligence, to help businesses and developers build, deploy, and manage applications through Microsoft's platform. [6].

The purpose of Microsoft Azure is to provide a flexible and scalable platform for hosting and managing applications and services. It allows developers to deploy applications without the need for on-site servers, reducing costs and overall complexities. Azure also supports services such as data analytics, machine learning, and security management, making it a well-rounded platform for various development needs.

Microsoft Azure was selected for its reliability, scalability, and wide range of services. Its ability to support multiple programming languages, frameworks, and tools makes it an excellent choice for development and deployment. Azure’s strong security features and GDPR compliance ensure data safety, which is crucial when selecting a cloud solution. The team selected Azure due to familiarity with several members and ease of use.

## Developing the Application

The development of the application followed the standard stages of the Software Development Life Cycle: Planning, Design, Development, and Testing. Each stage played a critical role in ensuring the success of the project and the completion of the final product with the team’s objectives.

### Planning

The planning stage began with the creation of a detailed Project Planning Report (PPR), which outlined the project’s aims, functional and non-functional requirements, and key deliverables.

The report also included an initial project timeline and allocated roles for each team member based on our skill sets, ensuring we worked as efficiently as possible. A risk assessment was conducted to identify potential issues, such as team member availability and technical difficulties, or illness within the team, and plans were put in place to reduce the impact of these risks such as Discord calls if members of the team were unwell. The PPR was a crucial document for guiding the team through the project.

### Design

The design stage focused on both the visual and functional aspects of the website. Initial wireframes were created using Balsamiq to outline the structure and layout of each page. These wireframes were developed into detailed designs using Figma, where visual elements such as the colour scheme, fonts, and user interface components were decided upon. UML diagrams, including use case and activity diagrams, were developed to show user interactions and the logic behind each of the key features the team was developing, such as booking sessions and managing user accounts. A colour pallet was also created to ensure consistency throughout the website.

### Development

Development was divided into front-end and back-end tasks, with team members specialising in their chosen areas.

* **Front-End Development:**

The front-end team used HTML, CSS, JavaScript, and Bootstrap to build a responsive and visually appealing user interface. The team used Bootstrap’s grid system to streamline the layout process while using custom CSS for styling. The team developed and implemented interactive features, such as form validations and navigation bars, using JavaScript. Regular commits and merges were made to a shared GitHub repository to ensure version control and encourage collaboration.

* **Back-End Development:**

The back-end team worked with Django and SQL to manage the server-side and database functionality, with Microsoft Azure providing a reliable and scalable platform for hosting the database. Django was used to speed up data handling and SQL managed the structure of the database.

The integration of the back end with the front end provided significant challenges for the team, and the systems could not be successfully connected by the end of the project. Despite great efforts to debug and resolve these compatibility issues, the team was unable to achieve a functional integration due to time restrictions. As a result of this, storing and fetching user information was not possible. This highlighted the importance of early integration testing and better collaboration between the front and back-end teams.

### Testing

Testing played a crucial role throughout the development process, ensuring the application’s functionality and performance met the project’s requirements. Front-end developers carried out unit and integration testing after each sprint to ensure that features were implemented correctly and met user requirements. Testing was performed using multiple browsers, including Google Chrome and Microsoft Edge, to ensure responsiveness and Compatibility across devices of different screen sizes. The back-end team carried out the database and server-side testing to confirm that queries functioned correctly. Once the main functionality was in place, user acceptance testing was performed, and several group members interacted with the application from a different user perspective. This approach helped identify and resolve usability issues, ensuring the application was easy to use and reliable for end users.

# Functional Testing

To determine the project's success, the group carried out functional testing to evaluate whether the criteria outlined in the project planning document had been achieved. Throughout the development lifecycle, three key testing methods were used: unit testing during development sprints, integration testing when merging components, and user acceptance testing after the development was completed. These methods were selected to ensure a full evaluation of the project and to identify any issues as early as possible.

## ****Unit Testing****

Unit testing focuses on individual components of the application and is crucial for quickly identifying and resolving bugs during development. This testing method was effective for the project due to usage at the end of each sprint. Each of the front-end developers was responsible for testing their code before finishing a sprint, ensuring that everyone was held accountable. This included features like form validation, navigation bar responsiveness, and displaying page elements. Back-end developers carried out unit tests on SQL queries and Django to ensure data manipulation was handled correctly. By isolating components, unit testing ensured that each piece of functionality worked as intended before moving on to the next phase.

## Integration Testing

Integration testing is used to ensure the interaction of different components within the website. This testing was carried out whenever front-end and back-end features were combined. It was important to identify issues such as data not showing correctly on the front end or server requests failing. During the development stage, the team used Git’s push and merge commands to add changes in increments, testing the integration of new components as they were added. There were significant challenges during this stage due to compatibility issues between the front and back end, which prevented successful integration in the end. Despite this, integration testing showed key areas for improvement during the project and ensured components functioned as expected.

## User Acceptance Testing

User acceptance testing (UAT) was conducted at the end of the development cycle to access the application from the perspective of an end user. This method is designed to confirm that the final product meets the user's requirements and provides a smooth experience. Several group members acted as test users, using the website features such as account creation, course browsing and filtering, and general navigation. Due to the incomplete integration, UAT was limited to the functionality of the front end. Pages were tested for responsiveness, simplicity, and ease of use across multiple devices and browsers. These tests provided the team with valuable feedback on the user experience and confirmed the front-end design met the project objectives.

# Review of Final Deliverable

The integration of the back end with the front end caused significant problems for the team, and the two systems could not be connected successfully by the end of the project. Despite a large amount of debugging to fix these issues, the time restrictions prevented the full integration. Due to this, storing and fetching user information was not possible. This highlighted the importance of early integration testing and better collaboration between the front and back-end teams.

The final deliverables show the potential of Tutord, and how it could be developed into a fully functional skill-sharing platform given more time. They also highlight several strengths and areas for improvement.

**Strengths:**

* A responsive and accessible front-end design provided a smooth user experience.
* Features such as course pages, navigation, and profile designs matched the project’s objectives and demonstrated the platform’s usability.

**Weaknesses:**

* The lack of integration between the front-end and back-end prevented the platform from fully functioning and showing the advanced features.
* The failed integration highlights the need for prioritising collaboration and early testing.

## Navigation

The navigation for the website was implemented using a fixed navbar at the top of each page. The navbar features a dropdown menu when viewed on mobile devices, ensuring the site is responsive across various screen sizes. The navigation bar contains links to all the key pages, a clickable logo that redirects users to the homepage, and a highlight feature to indicate the current page. While the wireframes didn’t include features like the dropdown menu, these were added to improve the user experience and accessibility. The navigation design meets its goal of ease of use and simplicity, providing the users with an efficient way to explore the website (shown in **9.3** and **9.4**).

## Homepage

The homepage introduces users to the purpose of the platform, showing its main features and benefits. It shows an engaging moving banner that displays different courses available on the site. The homepage design followed a similar layout to the wireframes but was enhanced to be more dynamic, using Bootstrap components. The homepage achieved its intended purpose of providing users with a welcoming introduction to the site (shown in **9.3**).

## Courses Page

The courses page features a grid layout showing available courses, including the name, tutor, rating, difficulty, and time commitment. A search bar at the top allows users to filter courses based on keywords. The page follows a similar design to the wireframes, but due to the unachieved integration with the back-end, dynamic interactions with the page were not possible. While the page offers users a preview of the courses, its functionality is limited by the lack of back-end integration (shown in **9.4**, **9.9**, and **9.10**).

## Sign-Up and Sign-In Pages

The sign-up and sign-in pages were designed to be easily accessible and simple. Both pages have a centred form with clear labels for each input field. A dropdown menu allows users to select their role (Tutor or Learner) during the sign-up. The layout and functionality align with the wireframes and Figma designs, full functionality such as connecting users to the database was not implemented due to integration issues. The front-end design remains simple and easy to use, allowing for improvements in the future (shown in **9.1** and **9.2**).

# Conclusions

**Project Process Review:**

The development of Tutord followed the key stages of the Software Development lifecycle: planning, design, development, and testing. The planning stage was crucial for establishing clear objectives, functional and non-functional requirements, and risk assessments. This ensured the team remained focused on developing a responsive and user-friendly web application. During the design stage, the team created a consistent style guide and visually appealing layouts for each of the pages, which the front-end team used for development.

The front-end team developed a variety of features, including course pages, responsive navigation bars, filtering for courses, and profile layouts, using HTML, CSS, JavaScript, and Bootstrap. The integration of the back-end, developed using Django and SQL, faces several technical challenges. These issues prevented the front end from interacting with the database, limiting the website's functionality. Despite these setbacks, the project displayed the team’s ability to work through challenges and deliver a strong prototype, that could be developed into a fully functioning site given more time.

**What Could Be Done Differently:**

If allowed to carry out this project again, the team would prioritise greater collaboration between the front and back-end developers, with a focus on early integration testing. Starting the integration testing earlier in the development cycle could have identified the issues early on and allowed developers to fix the problems before progress was made on certain components. Also, the team could have allocated more time to testing and debugging the database to ensure functionality. [2].

More use of project management tools like Trello could have improved organisation and task allocation. A Project Manager role could have also been assigned to someone, to ensure each member was on track and aware of tasks and responsibilities. Code reviews between the front-end team would have been useful as well, to ensure a high standard of code.

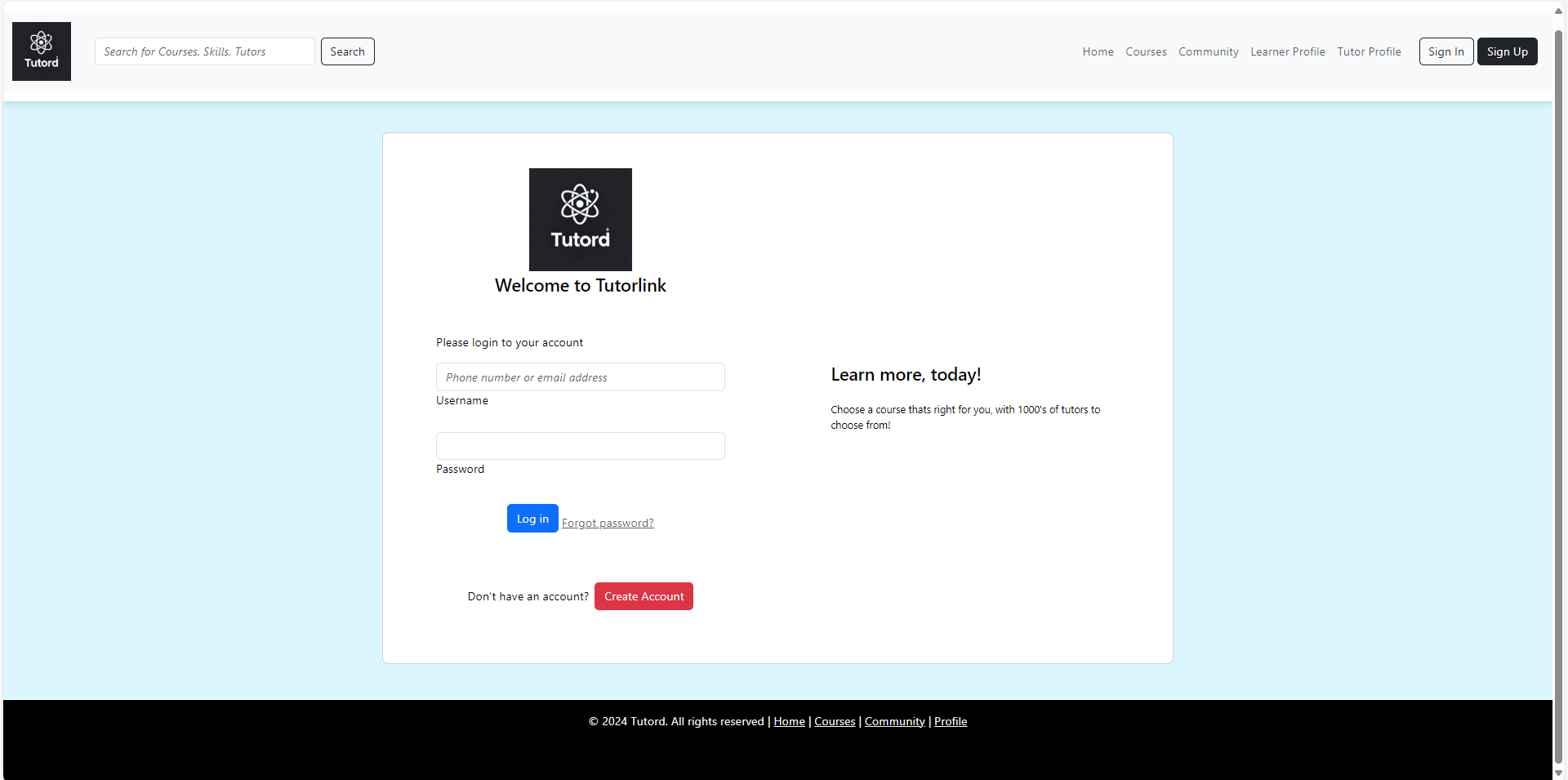
**Advice for Future Students:**

1. **Prioritise Early Integration:** Begin integration of the front and back end as soon as possible to identify and address any issues early in the development process.
2. **Allocate More Time for Testing:** Ensure plenty of time is set aside for detailed testing, with a focus on integration and user acceptance, to ensure all components function together
3. **Use Collaboration Tools Effectively:** Make use of tools like Trello and GitHub for task tracking and effective communication, reducing the risk of delays and duplication of work.

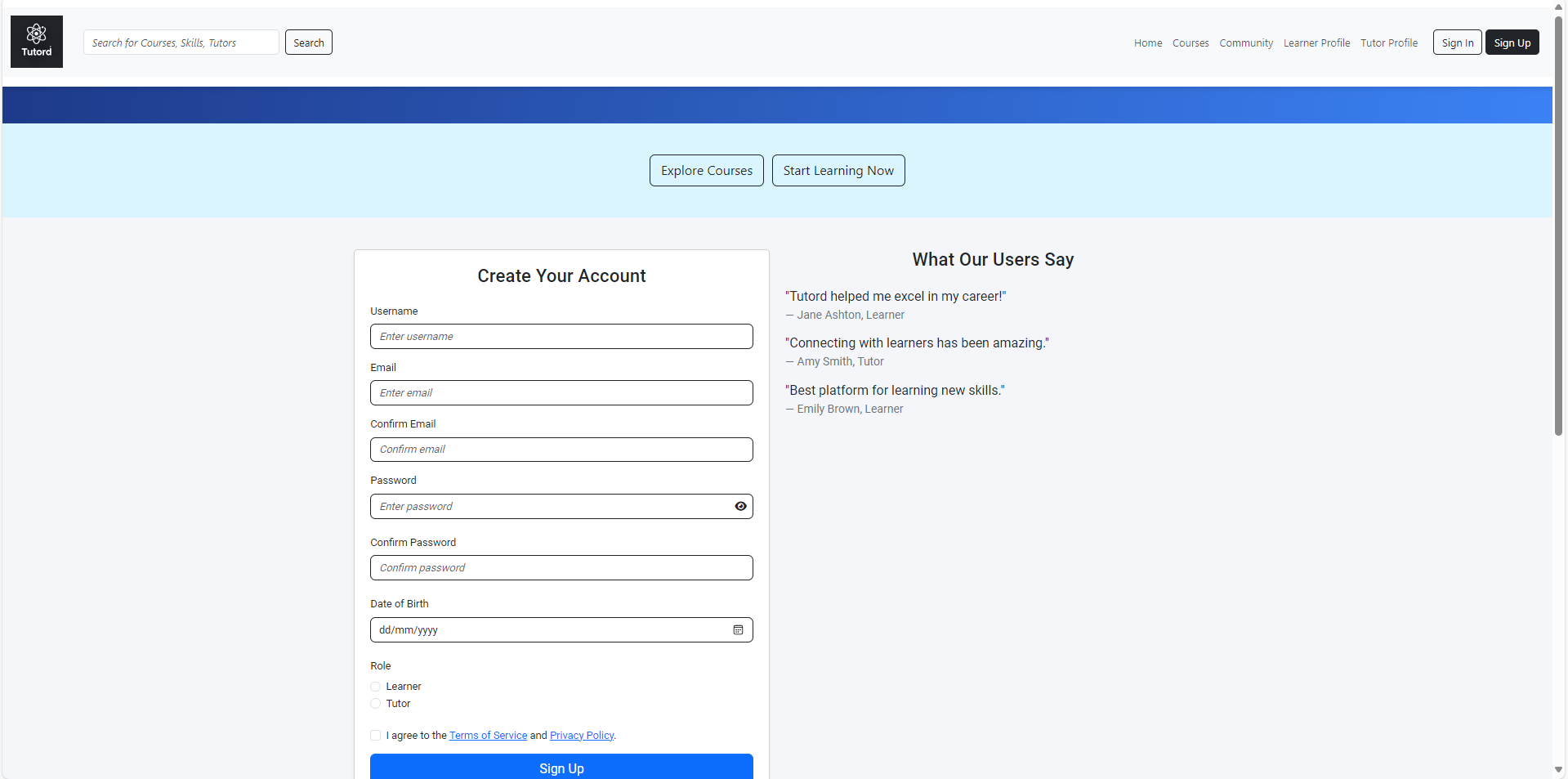
This project highlighted the importance of teamwork, time management, planning, and adaptability. Although the back-end integration limited the functionality, this experience provided a valuable look into the challenges faced in real-world software development projects and the importance of testing and integration.

# Appendix 1 – Product Screenshots

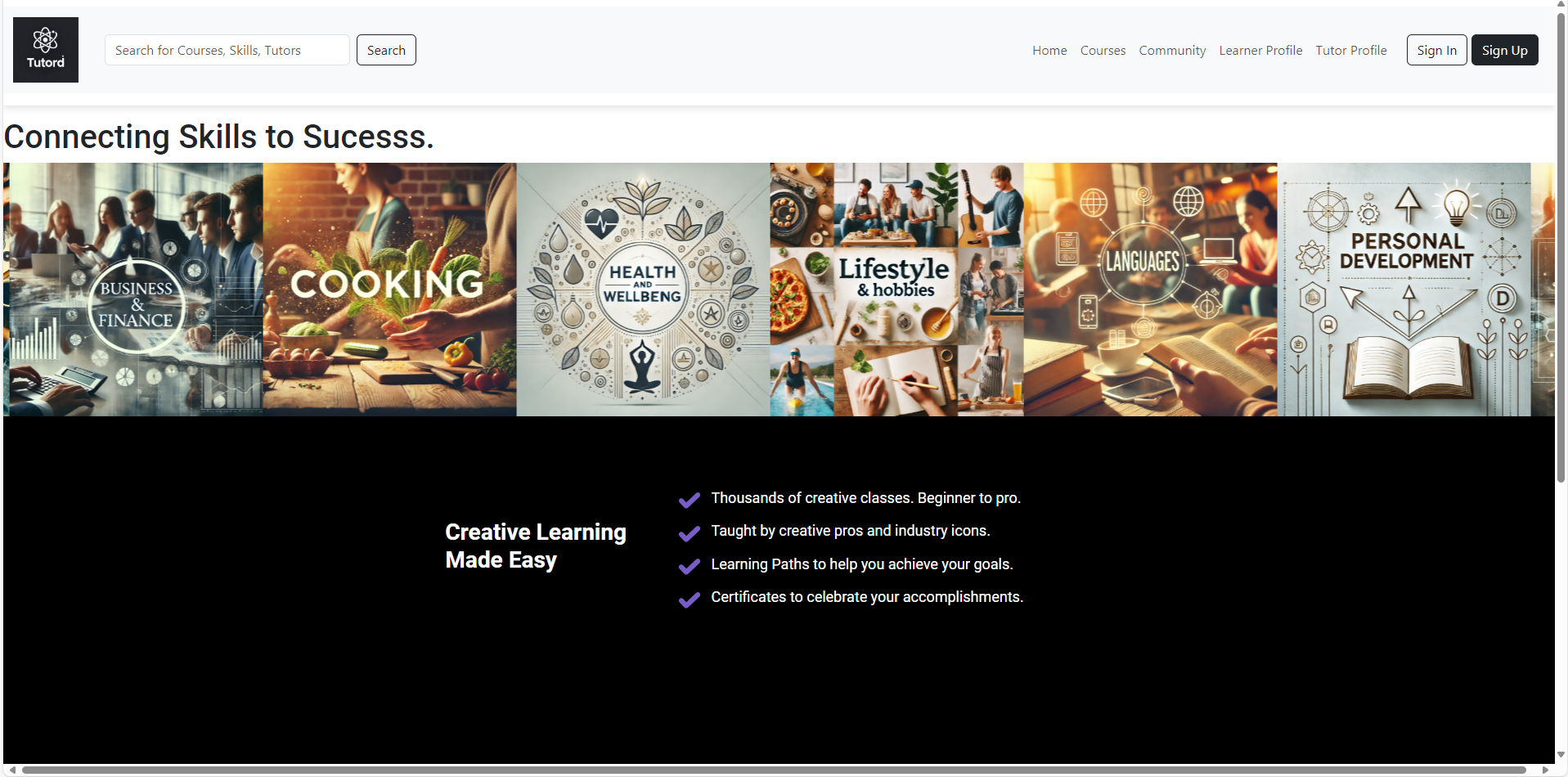
## Sign In



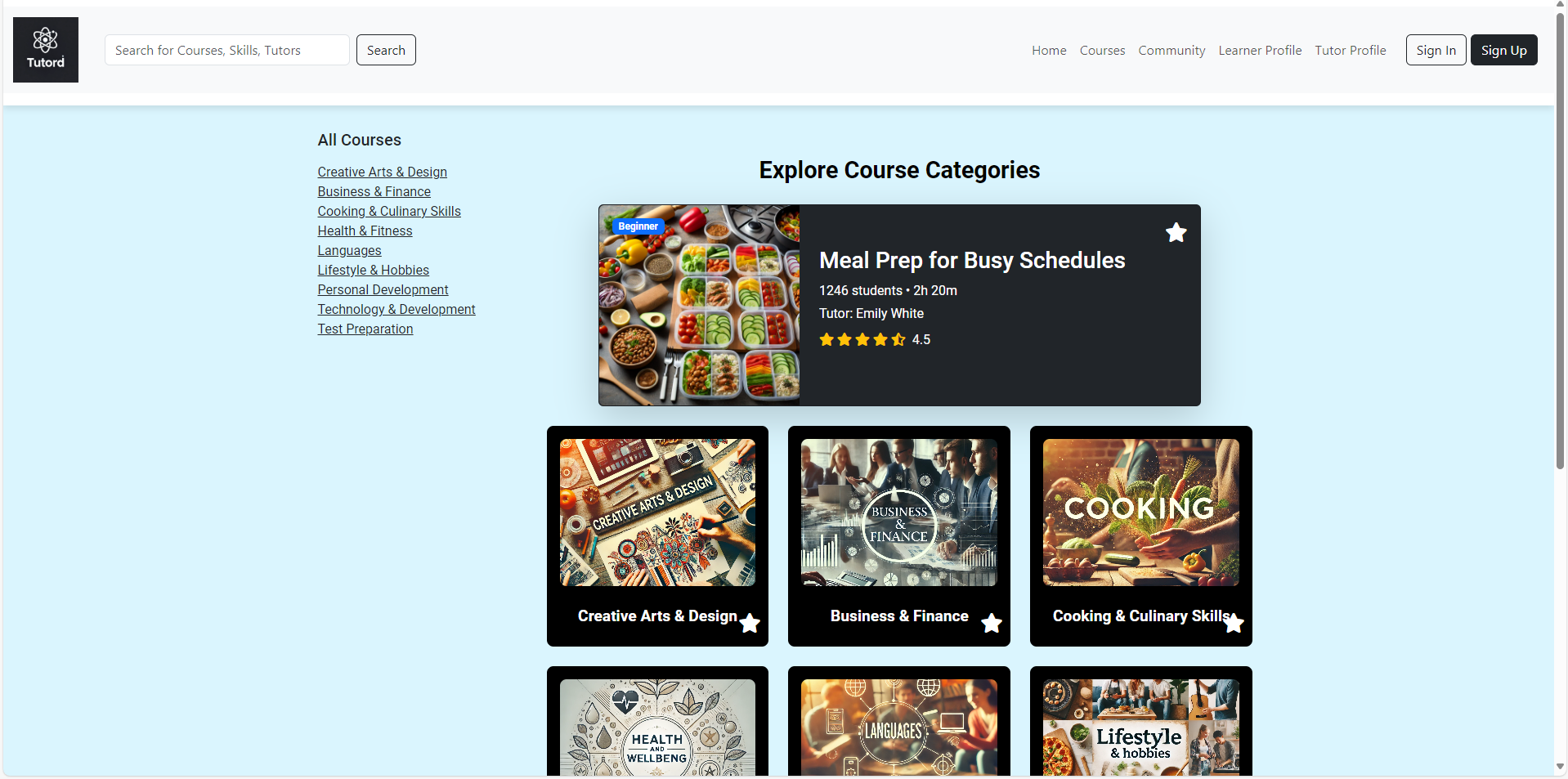
## Sign Up



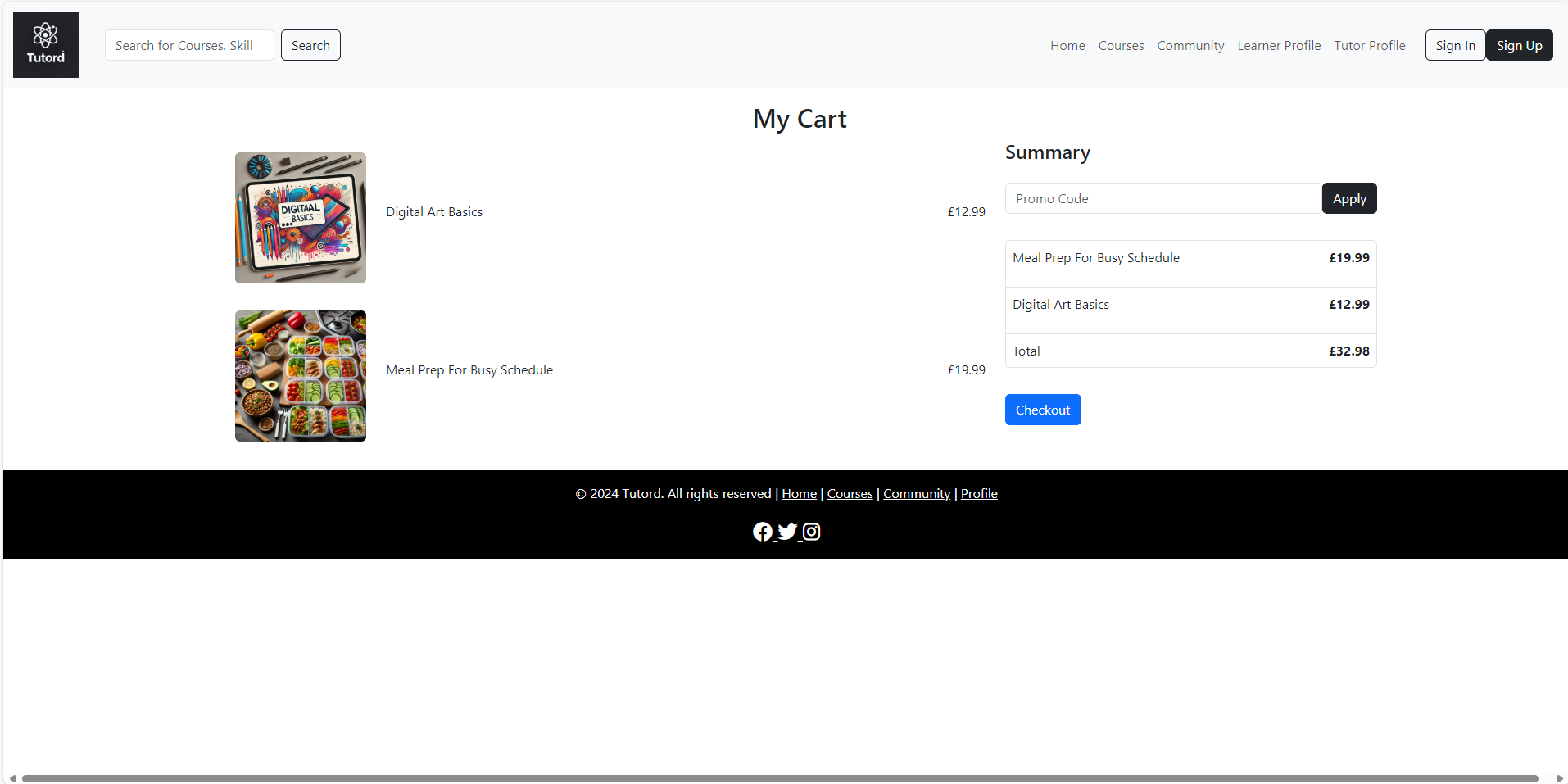
## Home Page



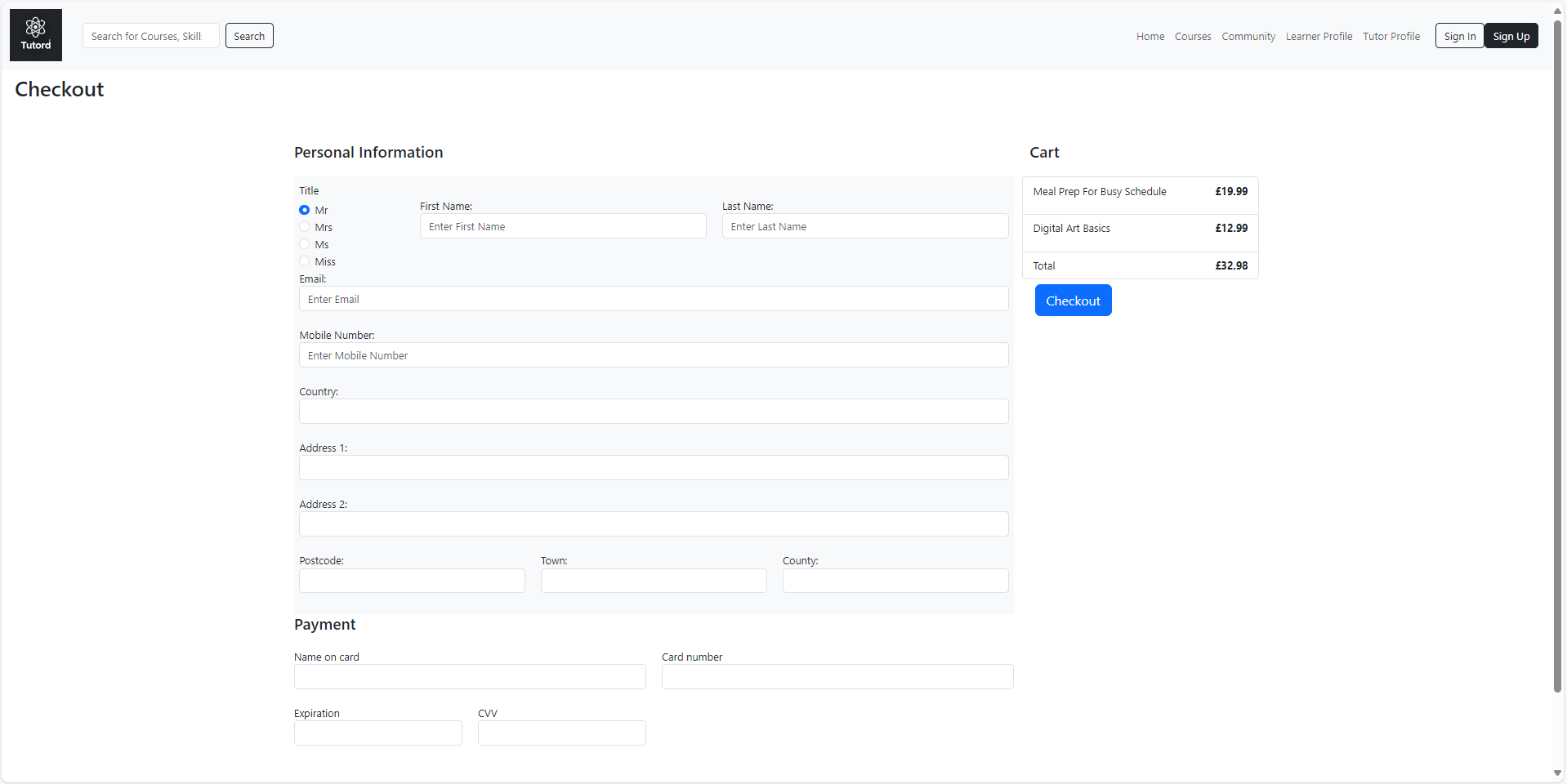
## Courses



## Cart



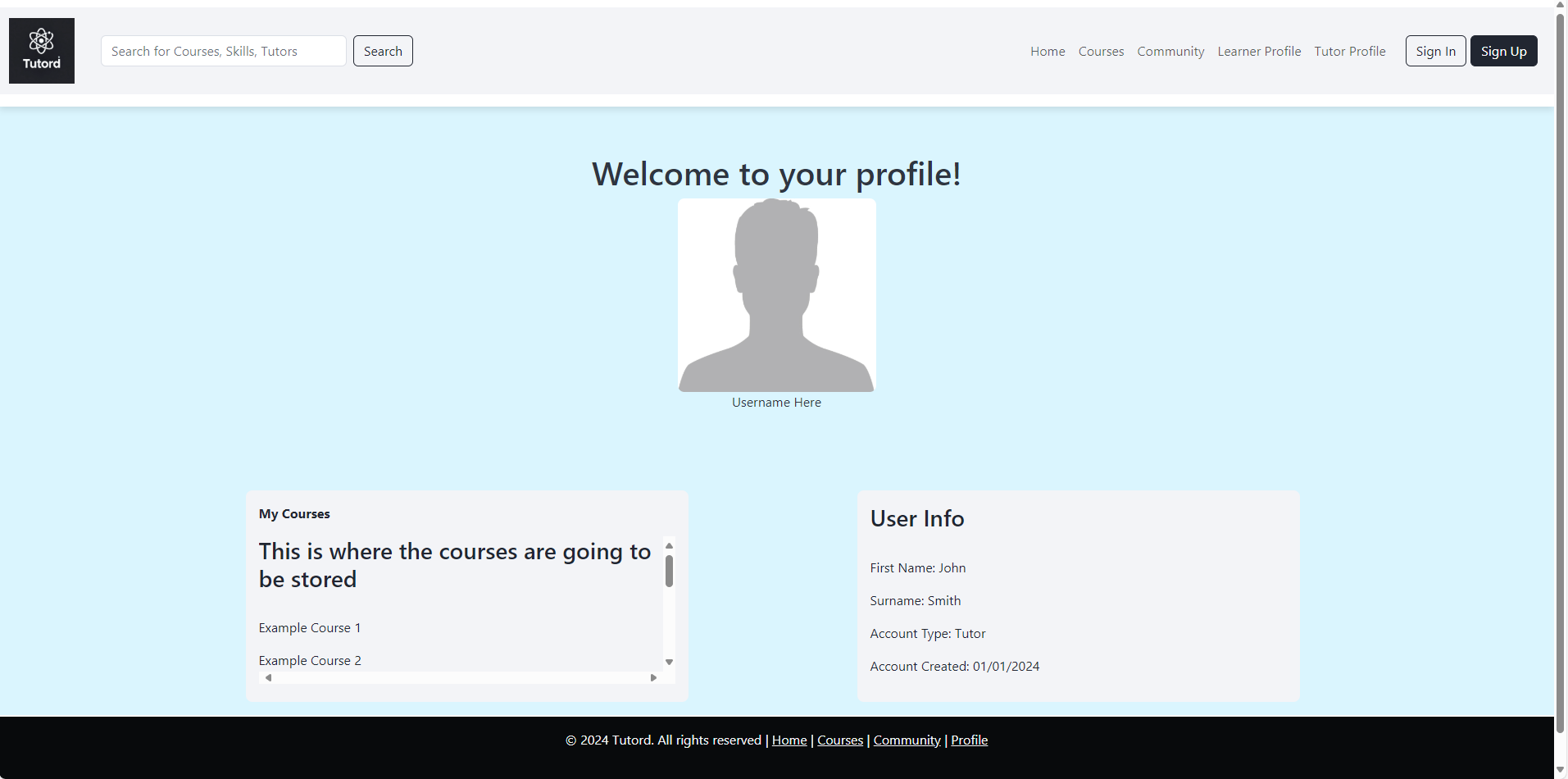
## Checkout



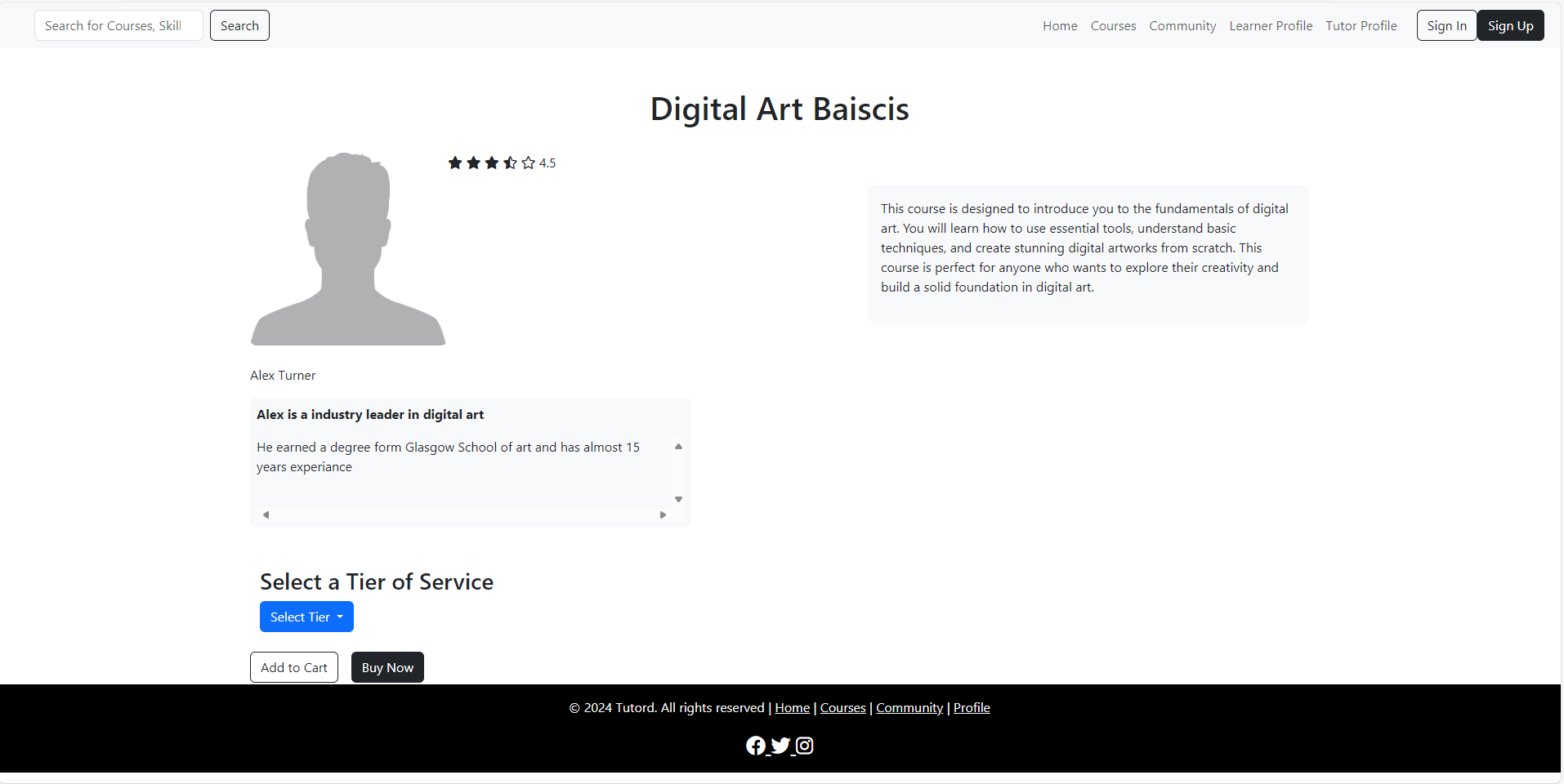
## Learner Profile Page



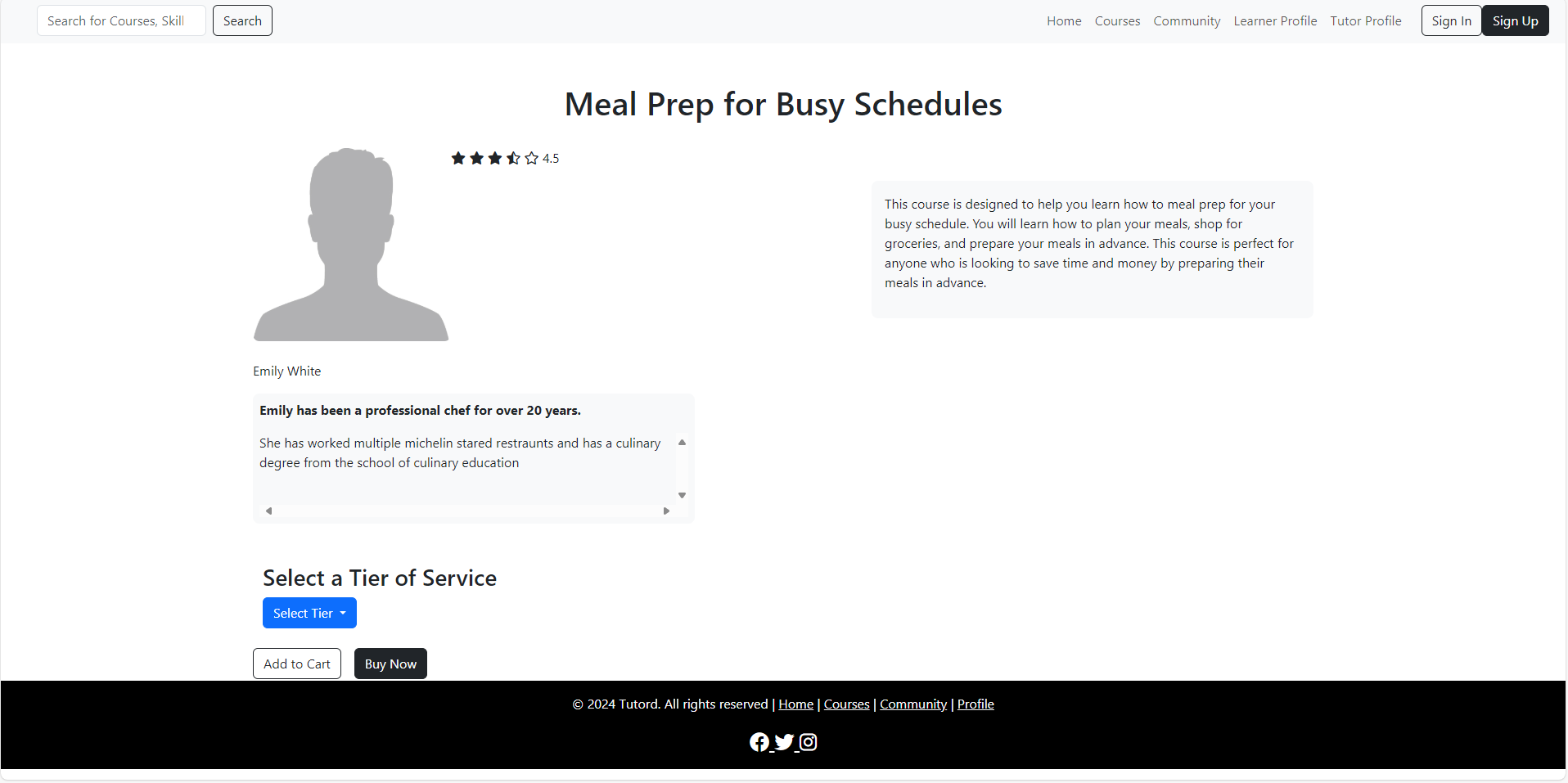
## Tutor Profile Page



## Digital Art Course Page



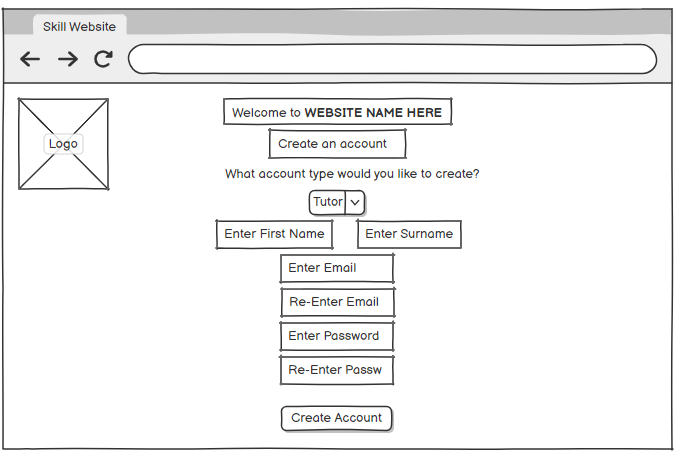
## Meal Prep Course Page



# Appendix 2 – Design Wireframes and Hi-Fidelity Examples

# 10.1 Desktop Wireframes

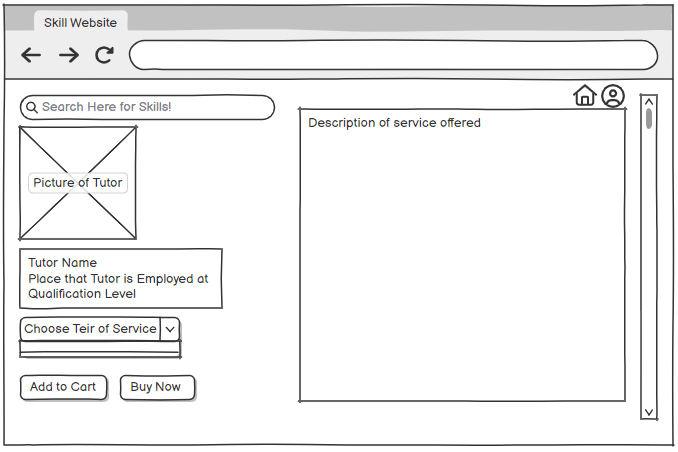
### Sign Up Page (Tutor)



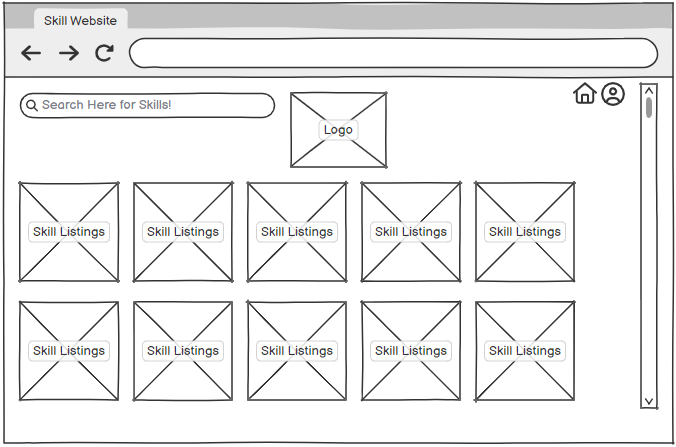
### Sign Up Page (Learner)



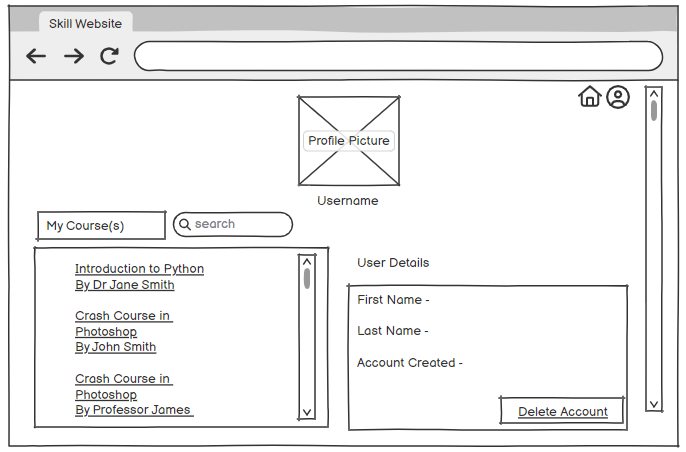
### Tutor profile



### Courses Page

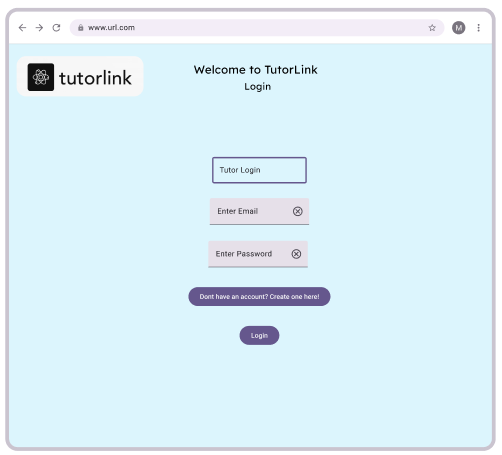


### Profile Page (Learner)

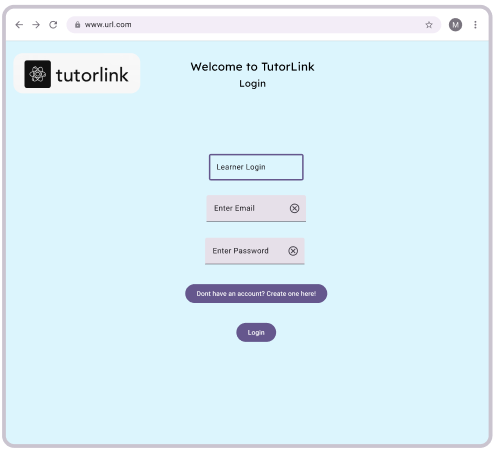


## Figma Interface Designs

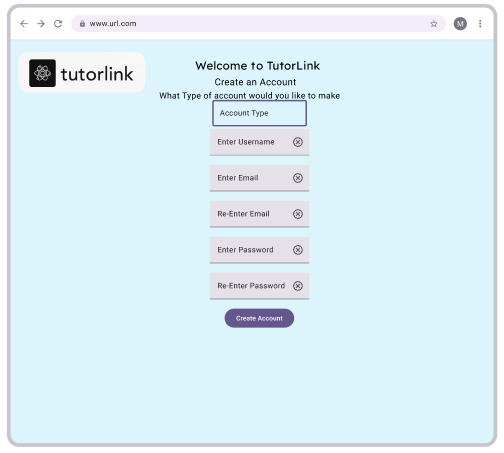
### Log In Page (Tutor)



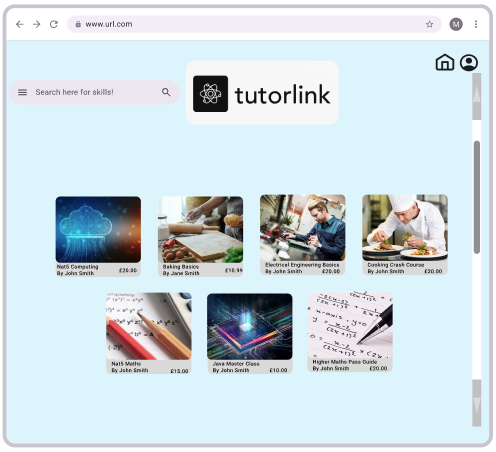
### Log In Page (Learner)



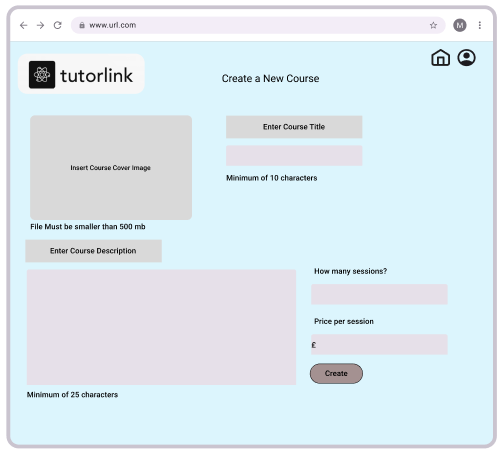
### Create Account Page



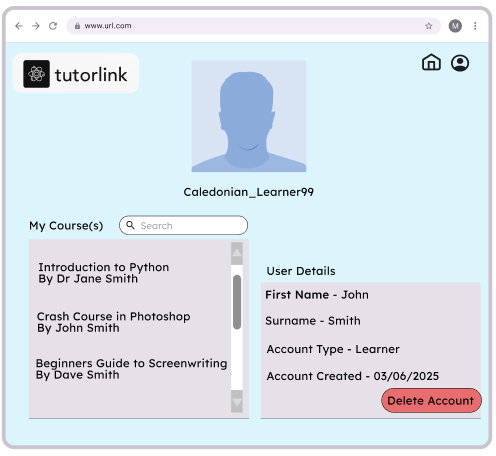
### courses Page



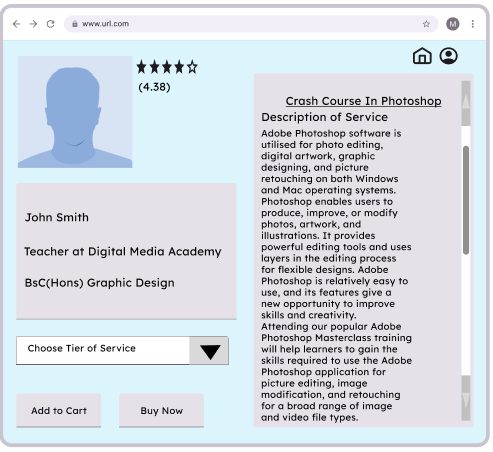
### Create Course Page



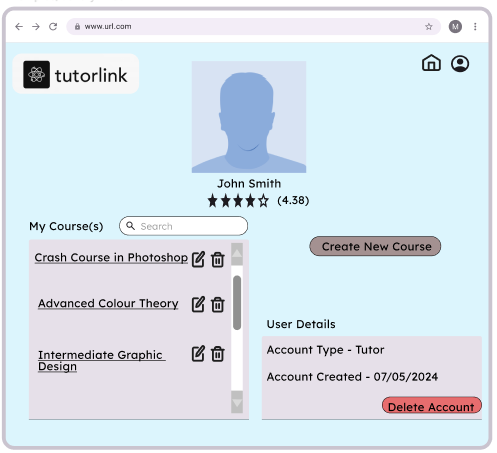
### Learner Profile Page



### Course Information Page



### Tutor Profile Page



# Appendix 3 Acceptance Tests

## Create, Sign-In, and Delete Account.

**Accounts**

|  |  |  |  |
| --- | --- | --- | --- |
| Title | User Story | Acceptance | Testing and working |
| Create Learner account | Create account with account type learner | Enter Email | Yes |
|  |  | Enter Password | Yes |
|  |  | Enter Name | Yes |
|  |  | Select account type as Learner | Yes |
|  |  | New account is added to database | No |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Title | User Story | Acceptance | Testing and working  (Yes/No/Part) |
| Sign in with account | Sign in with existing account | Enter Email | Yes |
|  |  | Enter Password | Yes |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Title | User Story | Acceptance | Testing and working |
| Delete account | Delete account that is currently signed in | Select “Delete Account” option | Yes |
|  |  | Any course this user had access to has -1 enrolled | Yes |
|  |  | Account is deleted from database | No |
|  |  |  |  |

## Courses Pages

|  |  |  |  |
| --- | --- | --- | --- |
| Title | User Story | Acceptance | Testing and working |
| Browse courses | Search, filter and select courses | Search for course name | Yes |
|  |  | Filter results sessions ascending | Yes |
|  |  | Select course | Yes |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Title | User Story | Acceptance | Testing and working  (Yes/No/Part) |
| Buy course - buy now | Buy course from course page | Select Tier of course | Yes |
|  |  | Select buy now | Yes |
|  |  | User is taken to checkout page | Yes |
|  |  | Enter Personal Details | Yes |
|  |  | Enter Card Details | Yes |
|  |  |  |  |

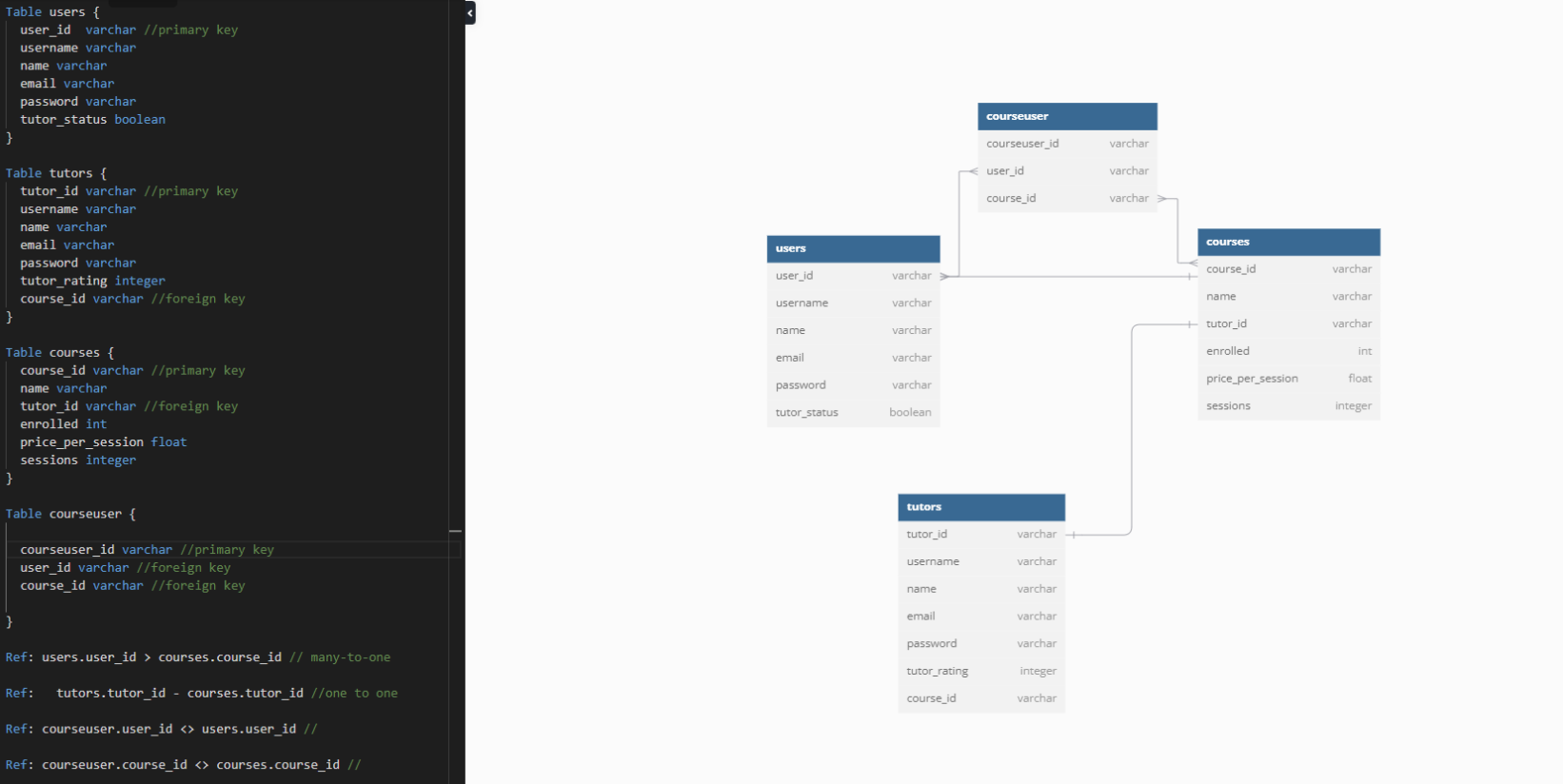
|  |  |  |  |
| --- | --- | --- | --- |
| Title | User Story | Acceptance | Testing and working |
| Buy course - cart | Buy course from cart page | Select Tier of course | Yes |
|  |  | Select add to cart | Yes |
|  |  | User is taken to checkout page | Yes |
|  |  | Enter Personal Details | Yes |
|  |  | Enter Card Details | Yes |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Title | User Story | Acceptance | Testing and working |
| Create Course | Create a new course as a Tutor | Enter course name | Yes |
|  |  | Enter number of sessions | Yes |
|  |  | Enter Price per Session | Yes |
|  |  | Enter course description | Yes |
|  |  | Course added to database | No |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Title | User Story | Acceptance | Testing and working |
| Alter Course | Alter an existing courses detail as a Tutor | Change course name | Yes |
|  |  | Change number of sessions | Yes |
|  |  | Change Price per Session | Yes |
|  |  | Change course description | Yes |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Title | User Story | Acceptance | Testing and working |
| Delete Course | Delete an existing course as a Tutor | Select course | Yes |
|  |  | Select “delete” option for chosen course | Yes |
|  |  |  |  |

# Entity Relationship Diagram



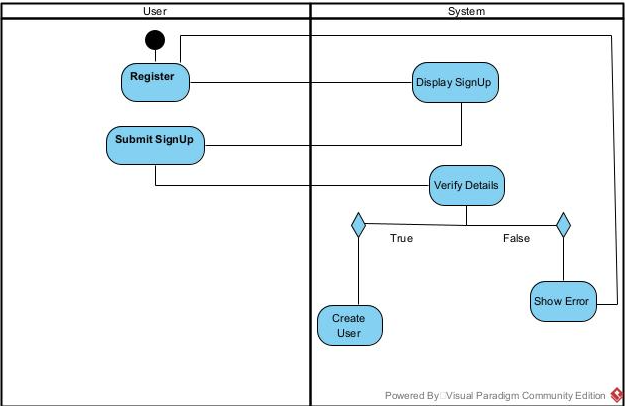
# Appendix 4 UML Diagrams

## Use Case Diagram

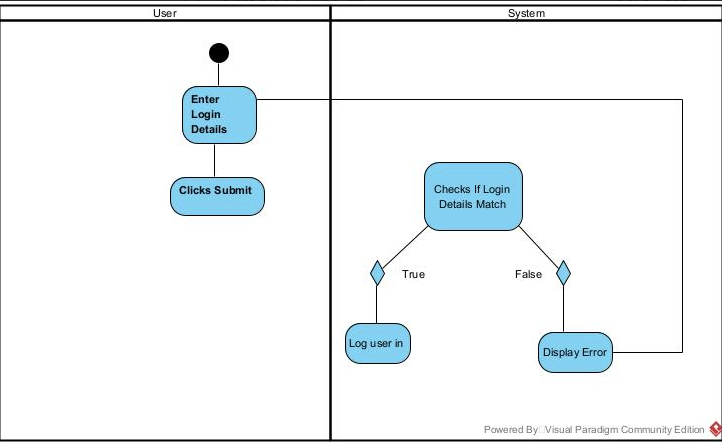


## Activity Diagrams

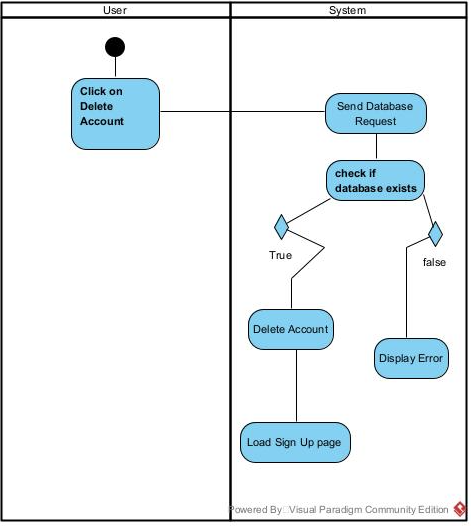
### Sign Up



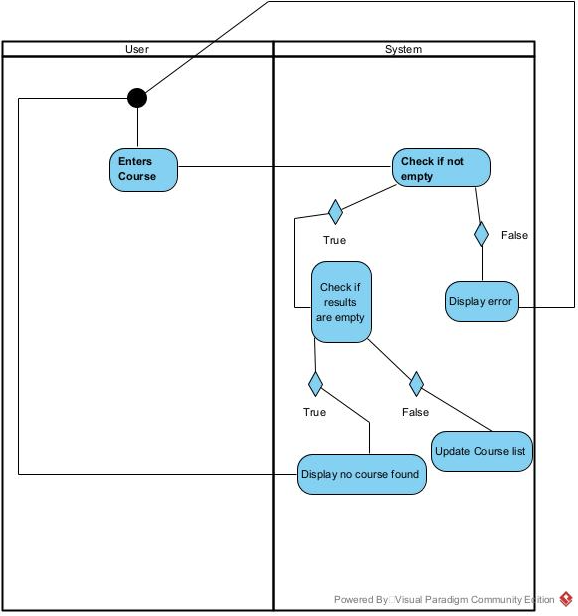
### Log In



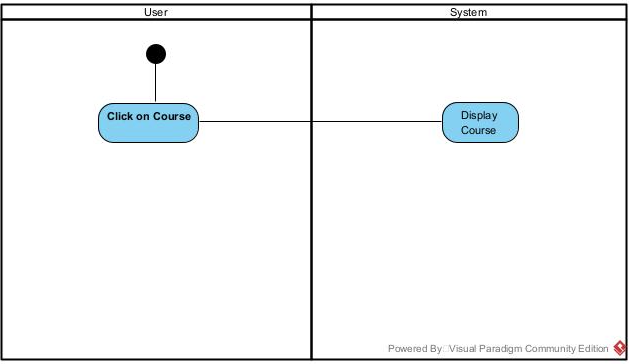
### Delete Account



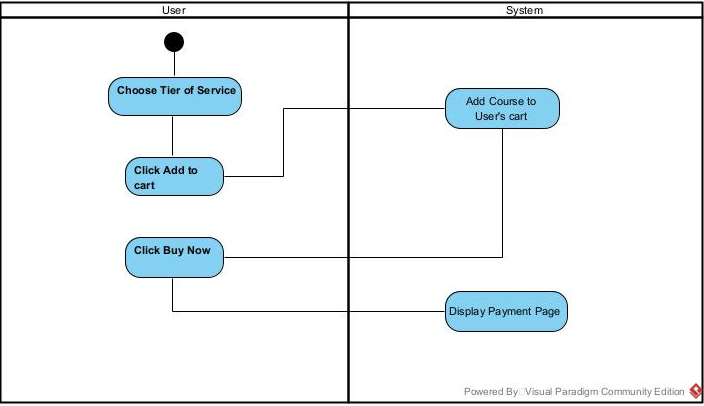
### Searching for Course



### Viewing Course



### Purchase Course



## 

# References

|  |  |
| --- | --- |
| [1] | “Getting Started with JavaScript,” Mozilla Developer Network, [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide. [Accessed 09 12 2024]. |
| [2] | “Web Content Accessibility Guidelines (WCAG) 2.1," 2018.,” World Wide Web Consortium (W3C), [Online]. Available: https://www.w3.org/TR/WCAG21/. [Accessed: 9-Dec-2024].. [Accessed 09 12 2024]. |
| [3] | “Introduction to HTML and CSS for Beginners,” Web Dev Academy, [Online]. Available: https://www.webdevacademy.com/html-css-guide/. [Accessed 09 12 2024]. |
| [4] | Python Cookbook, 3rd, ed. Sebastopol: O'Reilly Media, 2013. |
| [5] | “Mastering MySQL for Web Development,” *New York: Packt Publishing,* 2019. |
| [6] | “Microsoft Azure,” What is Microsoft Azure, [Online]. Available: https://azure.microsoft.com/en-us/overview/. [Accessed 09 12 2024]. |
| [7] | “Trello Blog,” Using Trello for Agile Teams, [Online]. Available: https://blog.trello.com/using-trello-for-agile/. [Accessed 09 12 2024]. |
| [8] | “Bootstrap,” Bootstrap Documentation, [Online]. Available: https://getbootstrap.com. [Accessed 09 12 2024]. |
| [9] | “Django Documentation,” Django, [Online]. Available: https://docs.djangoproject.com/. [Accessed 09 12 2024]. |
| [10] | “GitHub,” GitHub Guides, [Online]. Available: https://guides.github.com/. [Accessed 09 12 2024]. |