

Florida International University
Knight Foundation School of Computing and Information Sciences

Software Engineering Focus

Final Deliverable

Project Title: Learnr Progress and Skill Tracker

Team Members: Amani Hunt, Rose De Sicilia

Product Owner(s): Yola Belisario, Robert Fortunato

Mentor(s): Yola Belisario, Robert Fortunato, Rosamaria Belisario

Instructor: Masoud Sadjadi

The MIT License (MIT)
Copyright (c) 2016 Florida International University

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

ABSTRACT

The "Learnr Progress and Skill Tracker" aims to provide an effective solution for tracking the progress and skills of new professional technologists. This project was undertaken with a focus on agile development techniques and involved extensive planning, design, and testing.

At its core, the Learnr system facilitates the monitoring and management of learning pathways for mentees, mentors, and supermentors. The architecture of Learnr is divided into frontend and backend components, with the backend handling data management, server responses, and URL routing, and the frontend providing a user-friendly interface designed with Bootstrap.

The project's documentation provides a comprehensive view of the system's design, and includes user interface designs, sprint review reports, user manuals, and other relevant documents, offering a complete guide for users and developers alike.

In conclusion, this document serves as a comprehensive guide and knowledge transfer tool for future development teams.

TABLE OF CONTENTS

Introduction	8
• Current System	8
• Proposed System	8
 User Stories	10
• Implemented User Stories	10
• Pending User Stories	14
 Project Plan	14
• Hardware and Software Resources	15
• Sprints Plan	
○ Sprint 1	
○ Sprint 2	
○ Sprint 3	
○ Sprint 4	
○ Sprint 5	
○ Sprint 6	
○ Sprint 7	
 System Design	20
• Architectural Patterns	20
• System and Subsystem Decomposition	21
• Deployment Diagram	22
• Design Patterns	23

System Validation	26
Glossary	31
Appendix	32
• Appendix A UML Diagrams	32
• Appendix B User Interface Design	37
• Appendix C Sprint Review Reports	41
• Appendix D User Manuals, Installation/Maintenance Document, Shortcomings/Wishlist Document, and other documents	50
References	57

INTRODUCTION

Newcomers to the software engineering industry often struggle to find the personalized support they need to grow in their careers. The lack of accessible mentorship is a significant obstacle, and it can deter those with limited programming knowledge from pursuing software development.

To tackle this issue, our proposed Learnr platform aims to connect learners (mentees) with experienced educators (mentors and supermentors). In the dynamic world of software development, tailored guidance is crucial for mentees to thrive. This platform offers a set of user-friendly tools designed to provide personalized learning experiences and effective mentorship. The goal is to empower mentees to confidently navigate their educational journey, helping them overcome obstacles in their pursuit of a software development career.

Current System

When the Learnr project began, we started with a clean slate, as this was the first effort to bring this idea to life. The early stages involved gathering and organizing a wide array of feature requests from the Product Owners. This process helped in pinpointing the key elements needed for the system. The primary focus at this stage was to develop a Minimum Viable Product (MVP) that would capture the fundamental aspects of the platform, prioritizing practical functionality and the ability to adapt and evolve based on technical and design needs.

Proposed System

The Learnr Platform is a comprehensive solution designed for the Assurant Technology Development Pipeline to enhance their teaching and learning methodologies. The platform aims to bridge the gap between learners (mentees) and educators (mentors and supermentors), offering a suite of tools that support personalized learning and effective mentorship. This internal tool integrates features such as personalized learning roadmaps, account management, an intuitive user interface, secure database interactions, and security protocols.

The Learnr project uses a range of modern web technologies to create its Online Learning and Mentorship Platform. It employs Bootstrap, HTML, and CSS to make the user interface visually appealing and responsive. JavaScript adds interactivity, while

Express.js and Node.js handle the behind-the-scenes functionality. To improve data management, they transitioned from using CSV files to SQL. Security measures are in place to ensure that only authorized users can access sensitive information.

1. Learning Roadmap Management:

- This feature enables supermentors and mentors to create and manage custom learning roadmaps, which are central to the platform's educational approach. These roadmaps are directly linked to a comprehensive database, allowing for real-time updates and adjustments based on mentee progress.
- Mentors can assign, update, and track the progress of mentees through various learning pathways, ensuring that the educational content is tailored to each individual's needs and abilities.

2. Account Management and Personalization:

- The platform offers robust account management options. Users can edit their personal information, including passwords, email addresses, bios, school affiliations, and areas of interest. This personalization enhances the user experience and fosters a more engaging learning environment.
- Supermentors have extended privileges, such as registering new users, resetting passwords, and managing high-level account information. This administrative capability ensures smooth operation and maintenance of user data integrity.

3. User Interface and Experience (UI/UX):

- A significant focus has been placed on creating a user-friendly and aesthetically pleasing interface. Features include a sign-in interface, personalized user profiles, mentor and mentee dashboards, and management dashboards for overseeing the learning process.
- The UI/UX design is intuitive and responsive, catering to the needs of a diverse user base. It ensures easy navigation and access to key features, enhancing the overall usability of the platform.

4. Database Integration and Documentation:

- The backbone of the system is a well-structured database, which is meticulously documented, including an Entity-Relationship Diagram. This database efficiently associates mentors with mentees and manages the allocation and tracking of learning pathways.
- An automatic pathway ID assignment feature using hashing is implemented, adding a layer of efficiency and security to the management of learning content.

5. Security and Authorization:

- The security in this system incorporates authorization checks before granting access to sensitive pages and information. This approach protects user data and ensures that only authorized individuals have access to specific functionalities.

USER STORIES

The following section provides the detailed user stories that were implemented in this iteration of the Learnr project. These user stories served as the basis for the implementation of the project's features. This section also shows the user stories that are to be considered for future development.

Implemented User Stories

Account Management

1. Enable all users to sign in using a username and password.
2. Users with accounts created by supermentor can sign in with provided credentials.
3. Create the frontend that enables the super mentor to create user accounts (super mentor, mentor, mentee).
4. Allow supermentor to register another user after completing one registration.
5. Create a back end that enables super mentors to reset passwords for any account.
6. Create a front end that enables super mentors to reset passwords for any account.
7. Update backend to enable all users to sign in only using email and password.
8. Update front end to enable all users to sign in only using email and password.
9. Develop a login screen that maintains functionality and user experience.
10. Implement proper authorization checks before granting access to secure pages.
11. Update frontend of the super mentor and mentor management dashboards to adhere to our design principles.
12. Create a front end profile page that allows users to delete their own account.
13. Create backend logic that enables every user to delete their own account.
14. Develop a user profile following UI mockups.

UI/UX Design

1. Design mentee dashboard UI that shows progress within a single pathway.
2. Design UI side menu bar components for mentors, supermentors, and mentees.

3. Design a user profile UI.
4. Design mentor/supermentor dashboard UI that shows progress of all assigned mentees.
5. Design mentee dashboard UI that shows all assigned pathways.
6. Design a user-friendly sign-in interface using email and password.
7. Develop UI side menu bar components for mentors, supermentors, and mentees.
8. Build the front end for the mentor/supermentor dashboard UI that shows all their assigned mentees and their progress within their assigned pathways.
9. Build the front end for the mentee single pathway progress dashboard UI that shows progress within a single pathway.
10. Build the front end for the mentee dashboard UI that shows all assigned pathways.
11. Create a management dashboard for mentors.
12. Create a management dashboard for super mentors.

Learning Roadmap

1. Create the frontend that allows the supermentor to assign a learning pathway - CSV.
2. Create the backend that allows the supermentor to assign a learning pathway - CSV.
3. Create a front end that enables super mentors and mentors to assign existing learning roadmaps to a mentee.
4. Create a front end that enables super mentors and mentors to update mentee progress.
5. Create a front end that allows supermentors to enable previously disabled learning pathways.
6. Create a backend that allows supermentors to enable a previously disabled learning pathway.
7. Create a backend that allows supermentors and mentors to create learning roadmaps.
8. Create a backend that allows supermentors to disable learning pathways.
9. Create a backend that enables super mentors and mentors to assign existing learning roadmaps to a mentee.
10. Create a front end that allows supermentors and mentors to create learning roadmaps.
11. Create a backend that enables super mentors and mentors to update mentees' progress in pathways.
12. Allow supermentor and mentors to revisit the 'assign' page for already assigned pathways.
13. Create a front end that allows supermentors to disable learning pathways.

14. Create a front end that enables super mentors and mentors to assign existing learning roadmaps to a mentee.

Database Management

1. Reformat database from CSV to Relational Database to efficiently manage and access data.
2. Populate database with fictional users.
3. Create data for the database.
4. Create a database using an ER diagram.
5. Set up database connection files.
6. Develop an Entity-Relationship Diagram for the database.
7. Document the database schema.
8. Update database to allow pathways to be soft deleted.

Documentation and Research

1. Research technologies needed for MVP (JavaScript, HTML, CSS, Express.js, Node.js, NPM).
2. Create user stories for the MVP.
3. Categorize PO requests and decide on project MVP.
4. Identify parts of the MVP and associated technologies.
5. Discuss project needs with PO.
6. Create http://Readme.md file for the project.
7. Continuously update the project http://Readme.md file.

Software Design

1. Reformat codebase to follow an MVC architecture.

Workspace Setup

1. Set up GitHub desktop, GitHub repository, and Visual Studio Code.

Pending User Stories

Learning Roadmap

1. Enable super mentors and mentors to assign multiple learning roadmaps to a single mentee

2. "Allow supermentors and mentors to enhance learning roadmaps with external resources (links, documents, videos)"
3. Empower super mentors and mentors to assign coaches to mentees for skill development within their pathway
4. "Enable super mentors and mentors to create and save learning roadmaps for public or private use, detailing skills and acceptance criteria"
5. Enable super mentors and mentors to assign a chosen learning roadmap to multiple mentees
6. Enable super mentors and mentors to assign multiple learning roadmaps to multiple mentees
7. Enable super mentors and mentors to create and save a basic new learning roadmap outlining skills and acceptance criteria

Account Management

1. Provide users with a secure option to log out of their account
2. Add Ability to Amend Mentee Progress from Mentor/Supermentor Dashboard frontend
3. Add ability to amend user's name on profile frontend
4. Add ability to amend user's name on profile backend
5. Add Ability to Amend Mentee Progress from Mentor/Supermentor Dashboard backend
6. Enable all users to sign in using their Microsoft Credentials
7. Identify the additional abilities required by mentors
8. Identify the additional abilities required by admins
9. Identify the additional abilities required by mentees
10. Allow administrators to delete user accounts
11. Document the procedures for uploading content to GitHub
12. Allow supermentor to register another user after completing one registration
13. Enhance Logout Mechanism for Secure User Sessions
14. Enable administrators to designate mentors and mentees as coaches for skill sharing
15. Enable users to set up 2FA on their account

Community & Communication

1. Enable users to check their connection status on their profile's connection list
2. Facilitate user communication through messaging capabilities

3. Allow users to discover and view the existence of other users
4. Allow users to check connection status by viewing other users' profiles
5. Enable users to view incoming connection requests
6. Enable users to add comments or initiate forum discussions related to each skill or learning pathway
7. Facilitate users to initiate connections with others
8. Enable users to initiate connection requests with each other
9. Empower users to accept or reject connection requests
10. "Empower users to exchange attachments (images, documents) in messages"

Architecture

1. Design what happens after a pathway has been completed
2. Implement a way for mentor to be aware of mentees with no currently assigned pathways
3. Create architecture design for user connection and Badge Assignment Capabilities

Notifications

1. Allow mentees to opt-in for real-time notifications regarding pathway assignments
2. Facilitate real-time notifications for mentees when they earn badges
3. Offer mentees the option to receive daily reminders for their learning activities

Trophies and Rewards

1. Display mentees' badges and trophies on their account profile and dashboard
2. "Empower mentees to share their badges and trophies on social media platforms like LinkedIn, Twitter, etc."
3. Allow mentees to earn badges and trophies as they progress in learning pathways
4. Provide option for mentees' badges and trophies to be publicly or privately visible on their profile and dashboard

Software Design

1. Design Consistent Naming Conventions for Codebase
2. Enforce Consistent Naming Conventions for Codebase
3. Define Clear Version Control and Branching Strategies

UI/UX

1. Amend pages without design to follow design pattern
2. Ensure user-friendly mobile browser login with an intuitive design

PROJECT PLAN

This section describes the planning that went into the realization of this project. This project incorporated agile development techniques and as such required the sprints to be planned. These sprint plannings are detailed in the section. This section also describes the components, both software and hardware, chosen for this project.

Hardware and Software Resources

Hardware:

- None: The project does not require specific hardware.

Software:

- HTML, CSS, Javascript
- Bootstrap
- NPM
- Node.js
- Express.js
- Git
- GitHub and GitHub Desktop
- MySQL
- Visual Studio Code
- Figma
- StarUML
- Windows 11 PC OR Mac OS 13

Sprints Plan

Sprint 1

Description

In Sprint 1, the team focused on laying the groundwork for the "Learnr Progress and Skill Tracker" project. This included researching necessary technologies, setting up the development environment, and establishing the project's Minimum Viable Product (MVP). Key activities involved identifying and categorizing project requirements, discussing needs with Product Owners, and creating initial user stories. This foundational work was crucial for setting the direction and scope of the project.

User Stories

1. Research technologies needed for MVP (javascript, html, css, express.js, node.js, npm)
2. Create user stories for the MVP
3. Categorize PO requests and decide on project MVP
4. Identify parts of the MVP and associated technologies
5. Discuss project needs with PO
6. Set up github desktop, github repository, and vscode

Sprint 2

Description

Sprint 2 involved significant development in both the UI/UX and Learning Roadmap aspects of the project. The team designed various user interfaces, such as the mentee dashboard and the side menu bar components. Additionally, the backend functionalities for user account management and learning roadmap assignment were developed. The efforts in this sprint were directed towards building a cohesive and functional user experience.

User Stories

1. Design mentee dashboard UI that shows progress within a single pathway
2. Create the frontend that allows supermentor to assign a learning pathway CSV

3. Create the backend that allows supermentor to assign a learning pathway CSV
4. Design UI side menu bar components for mentors, supermentors, and mentees
5. Create front end that enables super mentor to create user accounts
6. Design a user profile UI
7. Design mentor/supermentor dashboard UI that shows progress of all assigned mentees
8. Design mentee dashboard UI that shows all assigned pathways
9. Allow supermentor to register another user after completing one registration
10. Design a user-friendly sign-in interface using email and password
11. Create backend that enables super mentor to create user accounts

Sprint 3

Description

In Sprint 3, the focus shifted towards refining the software design and enhancing account management features. The team restructured the code base to follow an MVC architecture, improving the scalability and maintainability of the software. Significant efforts were also made in creating functionalities for password management and documenting the project for ease of understanding and further development.

User Stories

1. Create http://Readme.md file for the project
2. Reformat code base to follow an MVC architecture
3. Create back end that enable super mentor to reset passwords for any account
4. Create front end that enable super mentor to reset passwords for any account

Sprint 4

Description

Sprint 4 was focused on integrating and enhancing account management with a SQL database, refining the database structure, and developing user interface components in alignment with the project's design principles. The transition to a relational database marked a significant improvement in data management and access. The team also concentrated on documenting the database schema and creating entity-relationship diagrams for better understanding.

User Stories

1. Create backend that allows users to view/edit/add their account information using SQL database
2. Reformat database from CSV to Relational Database
3. Populate database with fictional users
4. Create data for database
5. Set up database connection files
6. Develop Entity-Relationship Diagram for the database
7. Document the database schema
8. Update backend and front end to enable sign in using email and password

Sprint 5**Description**

The primary focus of Sprint 5 was on refining the account management system, particularly updating the backend and frontend for user authentication and authorization. This included developing a more user-friendly login screen and implementing security measures to restrict access to secure pages. The sprint aimed to enhance the overall user experience and security of the system.

User Stories

1. Update backend to enable all users to sign in only using email and password
2. Update front end to enable all users to sign in only using email and password
3. Develop a login screen that maintains functionality and user experience
4. Implement proper authorization checks before granting access to secure pages
5. Develop UI side menu bar components for mentors, supermentors, and mentees

Sprint 6**Description**

Sprint 6 continued to advance the Learning Roadmap capabilities of the project. The team worked on both frontend and backend aspects, focusing on enabling supermentors and mentors to manage learning roadmaps effectively. This included

functionalities for assigning, updating, and enabling/disabling pathways. The sprint also involved enhancing the user profile interface and database functionalities to support these new features.

User Stories

1. Update frontend of the super mentor and mentor management dashboards to adhere to design principles
2. Create front end that enables super mentors and mentors to assign existing learning roadmaps to a mentee
3. Create front end that enables super mentors and mentors to update mentee progress
4. Populate database with fictional users
5. Create data for database
6. Create front end that allows supermentors to enable previously disabled learning pathways
7. Create backend that allows supermentors to enable a previously disabled learning pathways
8. Create backend that allows supermentors and mentors to create learning roadmaps
9. Create backend that allows supermentors to disable learning pathways
10. Create backend that enables super mentors and mentors to assign existing learning roadmaps to a mentee
11. Continuously update the project <http://Readme.md> file
12. Create front end profile page that allows users to delete their own account
13. Build the front end for the mentor/supermentor dashboard UI that shows all their assigned mentees and their progress within their assigned pathways
14. Build the backend logic for the mentee single pathway tracker that gets info about progress within a single pathway
15. Build the backend logic for the mentor/supermentor dashboard UI that gets info about all assigned mentees and their progress within their assigned pathways
16. Build the front end for the mentee single pathway progress dashboard UI that shows progress within single pathway
17. Build the backend logic that gets info about all assigned pathway progress for the mentee dashboard
18. Build the front end for the mentee dashboard UI that shows all assigned pathways
19. Create front end that allows supermentors and mentors to create learning roadmaps
20. Create backend that enables super mentors and mentors to update mentees' progress in pathways
21. Create backend logic that enable every user to delete their own account

22. Allow supermentor and mentors to revisit the 'assign' page for already assigned pathways
23. Create front end that allows supermentors to disable learning pathways
24. Create front end that enables super mentors and mentors to assign existing learning roadmaps to a mentee
25. Update database to allow pathways to be soft deleted
26. Create a management dashboard for mentors
27. Create a management dashboard for super mentors
28. Develop a user profile following UI mockups

SYSTEM DESIGN

This section contains information on the design decisions that went into this project. The architecture patterns are outlined and explained. The entire system is shown in a package diagram and the subsystems are explained. Finally, the design patterns used in the project are discussed.

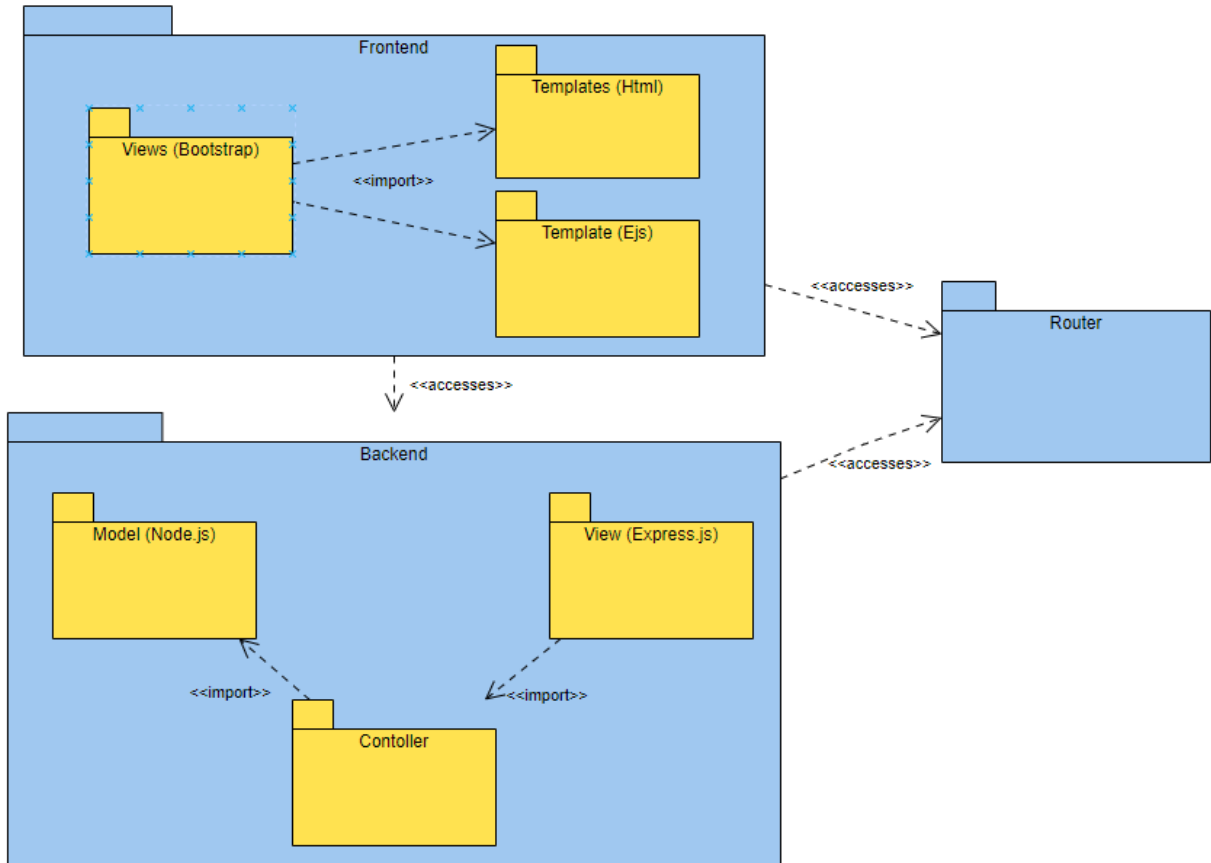
Architectural Patterns

Learnr's architectural design consists of two parts: the frontend and the backend.

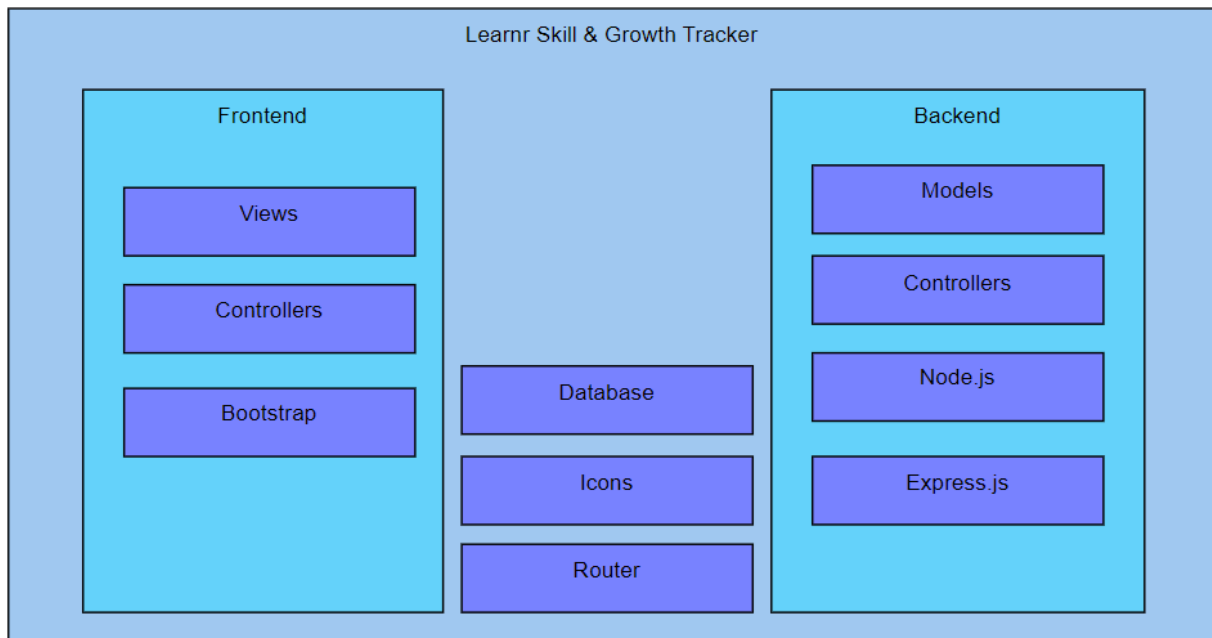
The backend can be broken down into 4 main components: the model, the view, the controller, and the router. The model is responsible for handling any kind of data within the MySQL database, while the view is responsible for sending either html files or ejs file to be rendered by the server. The controller is responsible for sending and receiving information between the model and view. The router controls url redirection/handling for the entire server. The backend uses a mix of Express.js and Node.js to perform server requests and responses. It also makes requests and receives responses from the frontend which allows it to access endpoints within the database to query any number of information.

The frontend is built using HTML, CSS, Javascript, and Bootstrap.

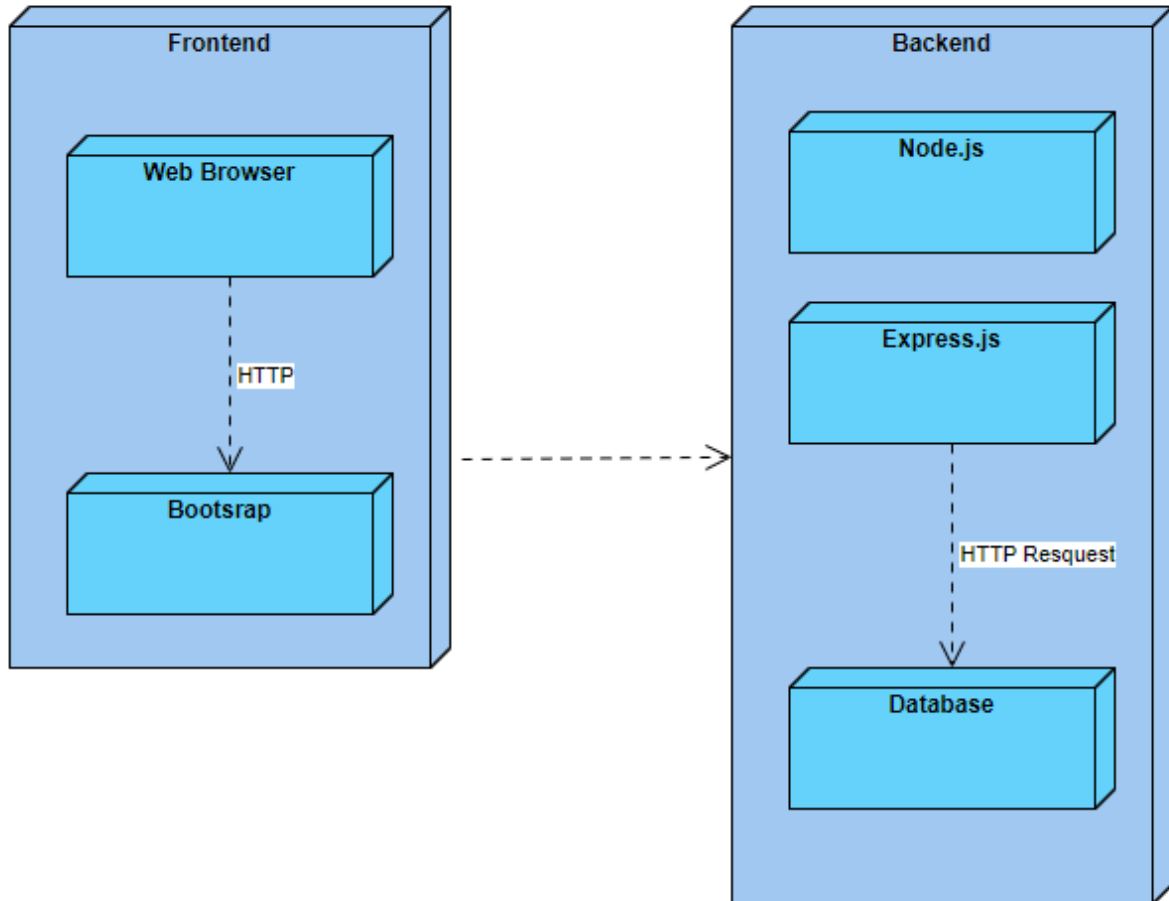
Package Diagram



System and Subsystem Decomposition

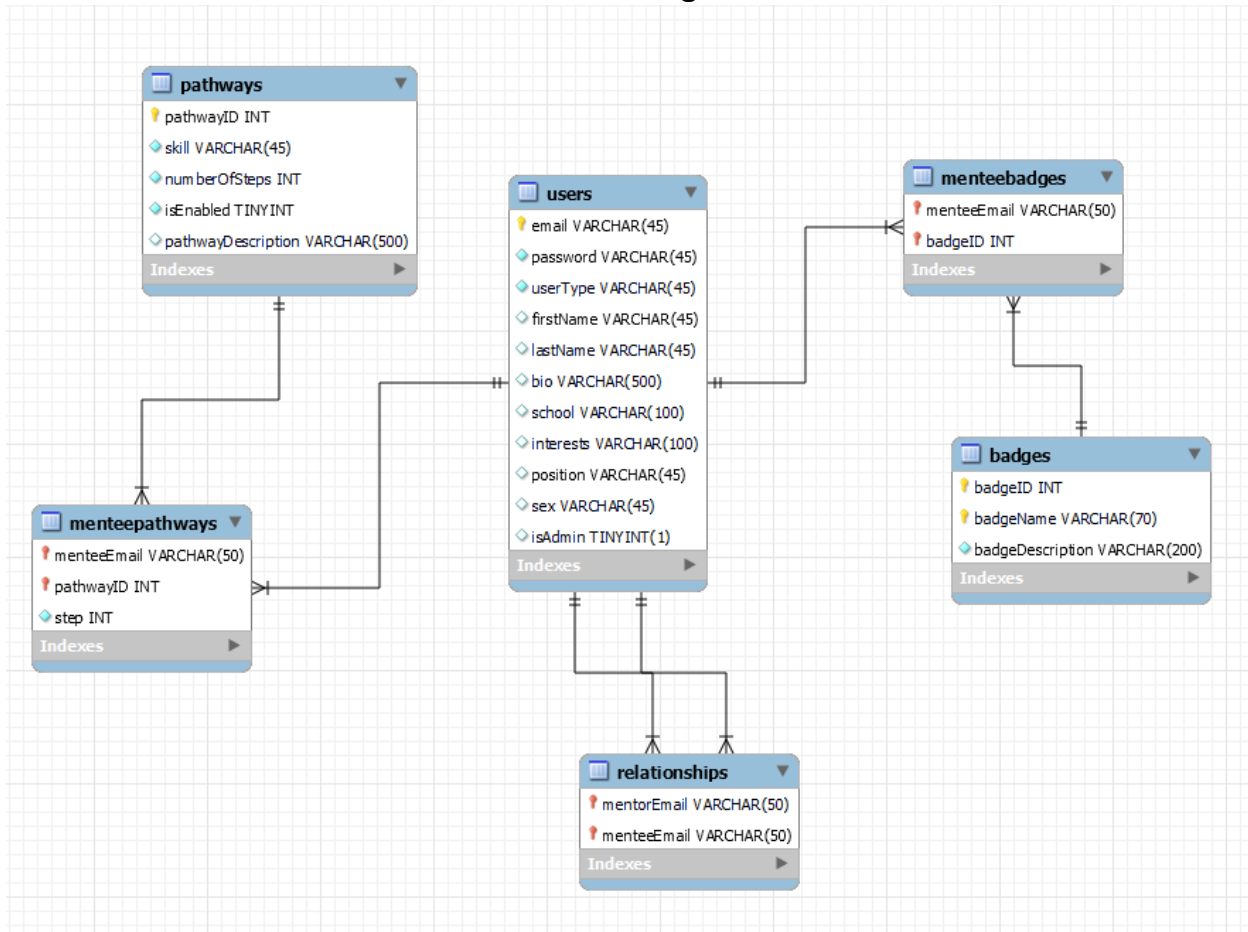


Deployment Diagram

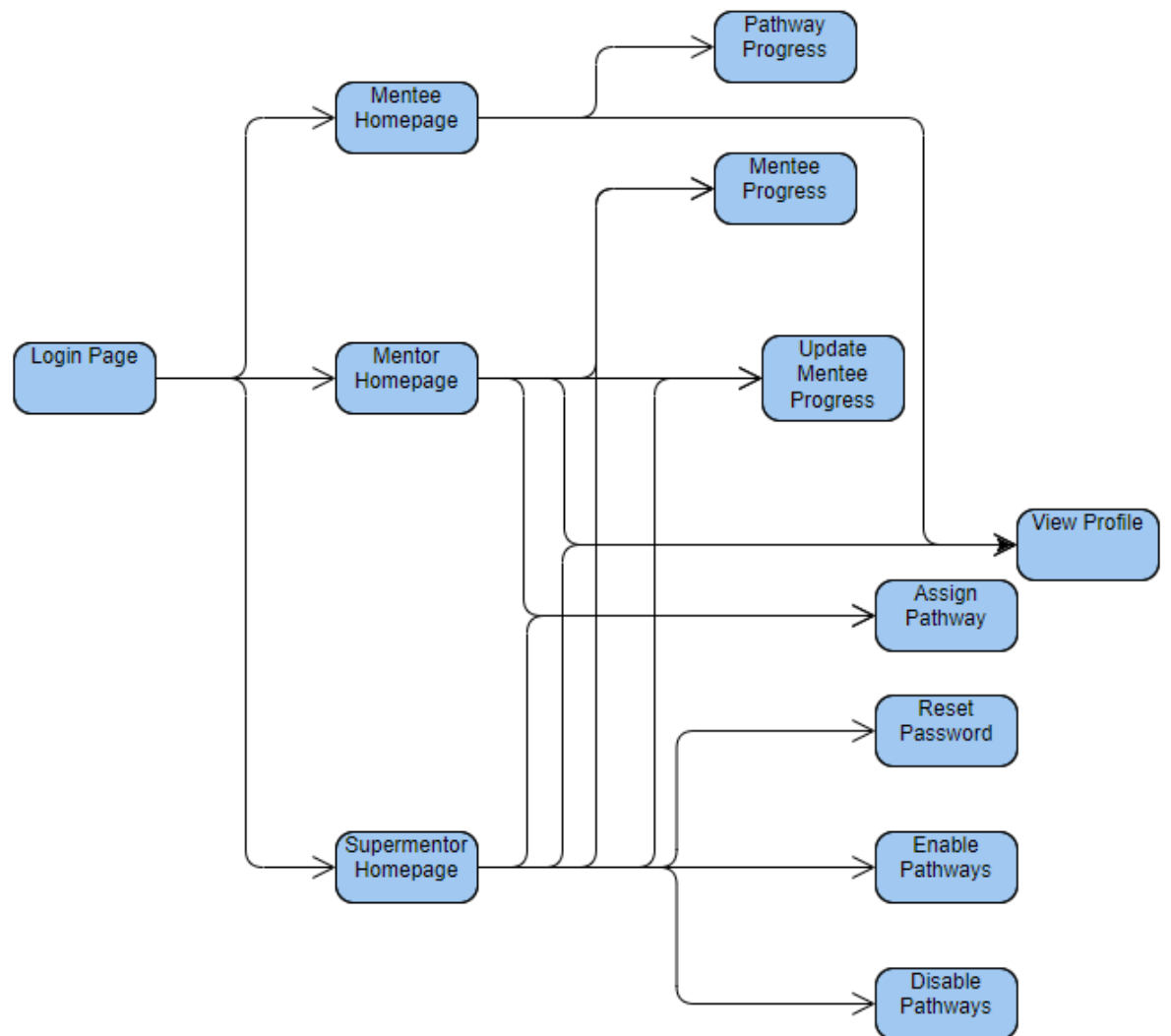


Design Patterns

Data Design



System Design



SYSTEM VALIDATION

In this section, test cases for all developed features are presented.

Capstone 2 Development

Test Learnr UI

Test Case ID: Display_Login

- **Purpose:** To display the Learnr login page.
- **Setup:** Run server and access localhost:3000 on a web browser.
- **Input:** Server.js file containing the basic GUI.
- **Expected Output:** Login GUI on the localhost:3000 server.
- **Status:** Pass

Test Learnr UI Profile Mentee

Test Case ID: Display_UI_Profile_Mentee

- **Purpose:** To display the Learnr mentee Profile page.
- **Setup:** Run server and access localhost:3000 on a web browser. Login with mentee credentials. Click on the profile tab on the left side of the screen.
- **Input:** Server.js file.
- **Expected Output:** Profile page renders with user's information.
- **Status:** Pass

Test Learnr_UI_Profile_Mentor**Test Case ID: Display_UI_Profile_Mentor**

- **Purpose:** To display the Learnr mentor Profile page.
- **Setup:** Run server and access localhost:3000 on a web browser. Login with mentor credentials. Click on the profile tab on the left side of the screen.
- **Input:** Server.js file.
- **Expected Output:** Profile page renders with user's information.
- **Status:** Pass

Test Learnr_UI_Profile_Mentee**Test Case ID:
Display_UI_Profile_Supermentor**

- **Purpose:** To display the Learnr mentee Profile page.
- **Setup:** Run server and access localhost:3000 on a web browser. Login with mentee credentials. Click on the profile tab on the left side of the screen.
- **Input:** Server.js file.
- **Expected Output:** Profile page renders with user's information.
- **Status:** Pass

Test Learnr UI Profile Supermentor**Test Case ID:****Display_UI_Profile_Supermentor**

- **Purpose:** To display the Learnr supermentor Profile page.
- **Setup:** Run server and access localhost:3000 on a web browser. Login with supermentor credentials. Click on the profile tab on the left side of the screen.
- **Input:** Server.js file.
- **Expected Output:** Profile page renders with user's information.
- **Status:** Pass

Test Learnr UI Mentee Pathway Tracker**Test Case ID: Display_UI_Mentee_Tracker**

- **Purpose:** To display the Learnr mentee tracker page.
- **Setup:** Run server and access localhost:3000 on a web browser. Login with mentor/supermentor credentials.
- **Input:** Server.js file.
- **Expected Output:** Home page renders with current mentee's progress.
- **Status:** Pass

Test Learnr_Assign_Pathway**Test Case ID: Pathway_Assign**

- **Purpose:** To display the assigned pathway on a mentee's home page.
- **Setup:** Run server and access localhost:3000 on a web browser. Login with supermentor credentials. Click the management tab on the left and select 'Assign a pathway'. Enter mentee and pathway information and click assign. Logout and log into the mentee's account.
- **Input:** Server.js file.
- **Expected Output:** Mentee's Home page shows the newly assigned pathway.
- **Status:** Pass

Test Learnr_Disable_Pathway**Test Case ID: Pathway_Disable**

- **Purpose:** To disable a pathway in the database and remove it from mentee's assigned pathways lists.
- **Setup:** Run server and access localhost:3000 on a web browser. Login with supermentor credentials. Click the management tab on the left and select "Disable a pathway". Select pathway to disable and select "Disable Pathway". Open the database.
- **Input:** Server.js file.
- **Expected Output:** Database updates with the selected pathway's 'isEnabled' value set to 0 and the corresponding mentee pathway was removed from the table.
- **Status:** Pass

Test Learnr_Enable_Pathway**Test Case ID: Pathway_Enable**

- **Purpose:** To Enable a currently disabled pathway in the database.
- **Setup:** Run server and access localhost:3000 on a web browser. Login with supermentor credentials. Click the management tab on the left and select "Enable a pathway". Select pathway to enable and select "Enable Pathway". Open the database.
- **Input:** Server.js file.
- **Expected Output:** Database updates with the selected pathway's 'isEnabled' value set to 1.
- **Status:** Pass

Test Learnr_Register_User**Test Case ID: Add_User**

- **Purpose:** To ensure user are correctly registered in the database.
- **Setup:** Run server and access localhost:3000 on a web browser. Login with supermentor credentials. Click the management tab on the left and select "Register New User". Enter new user credentials and click register. Open the database.
- **Input:** Server.js file.
- **Expected Output:** Database updates with the new user.
- **Status:** Pass

GLOSSARY

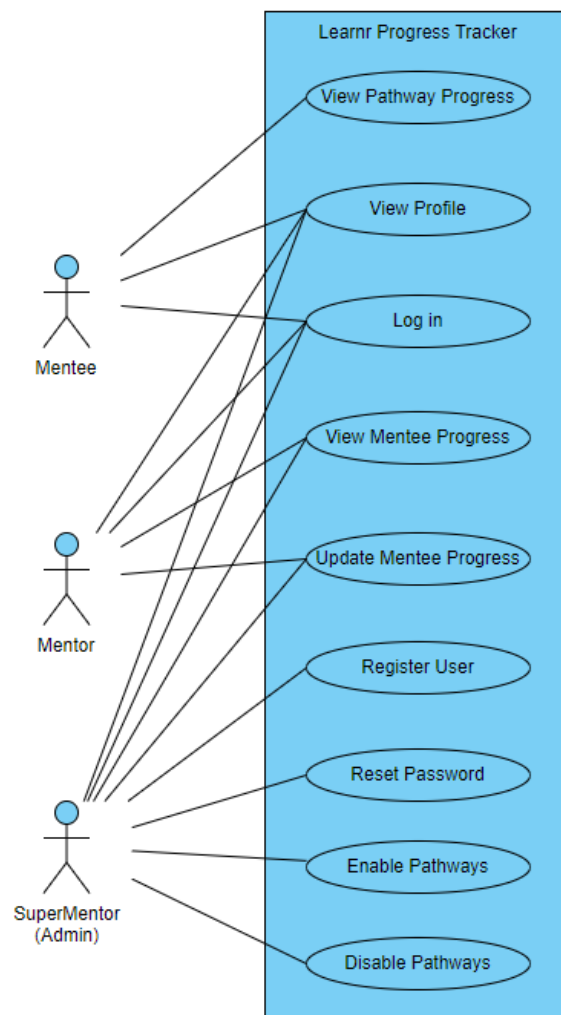
- Agile Development: A software development methodology emphasizing iterative development, collaboration, and adaptability throughout the project lifecycle.
- Bootstrap: A front-end framework used for designing websites and web applications with HTML, CSS, and JavaScript.
- CSV (Comma-Separated Values): A file format used to store tabular data, such as numbers and text, in plain text form.
- Express.js: A web application framework for Node.js, designed for building web applications and APIs.
- Figma: A vector graphics editor and prototyping tool primarily web-based.
- Git: A distributed version control system used for tracking changes in source code during software development.
- GitHub: A web-based hosting service for version control using Git, offering distributed version control and source code management functionality.
- HTML (Hypertext Markup Language): The standard markup language for documents designed to be displayed in a web browser.
- JavaScript: A programming language commonly used in web development for creating interactive effects within web browsers.
- Learning Roadmap: A predefined set of learning objectives and materials, forming a structured path for educational progression.
- Mentee: An individual receiving guidance and training, especially in a professional setting.
- Mentor: An experienced and trusted advisor who provides guidance and training to a less experienced or knowledgeable person (mentee).
- MVC (Model-View-Controller): A software design pattern commonly used for developing user interfaces that divide the related program logic into three interconnected elements.
- MySQL: An open-source relational database management system based on SQL (Structured Query Language).
- Node.js: An open-source, cross-platform, JavaScript runtime environment that executes JavaScript code outside of a browser.
- NPM (Node Package Manager): A package manager for the JavaScript programming language, default for Node.js.
- Relational Database: A digital database based on the relational model of data, which organizes data into one or more tables with columns and rows.

- SQL (Structured Query Language): A domain-specific language used in programming and designed for managing data held in a relational database management system.
- StarUML: A software modeling tool used for creating UML (Unified Modeling Language) diagrams.
- Supermentor: A highly experienced mentor providing guidance at a higher or more advanced level than typical mentors.
- UML (Unified Modeling Language): A standardized modeling language enabling developers to specify, visualize, construct, and document artifacts of software systems.
- User Story: A tool used in Agile software development to capture a description of a software feature from an end-user perspective.
- Visual Studio Code: A source-code editor made by Microsoft for Windows, Linux, and macOS, includes support for debugging, embedded Git control, syntax highlighting, intelligent code completion, and more.
- 2FA (Two-Factor Authentication): An extra layer of security used to ensure the security of online accounts beyond just a username and password.

APPENDIX

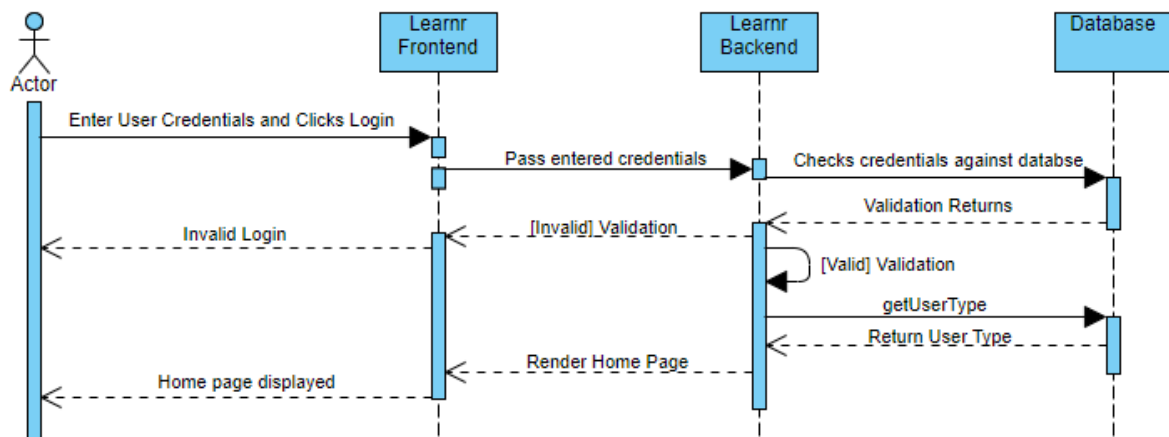
Appendix A: UML Diagrams

1. Use Case Diagrams

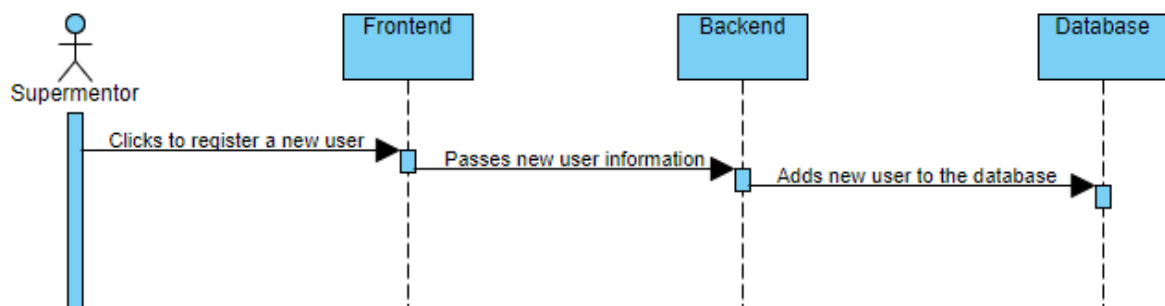


2. Sequence Diagrams

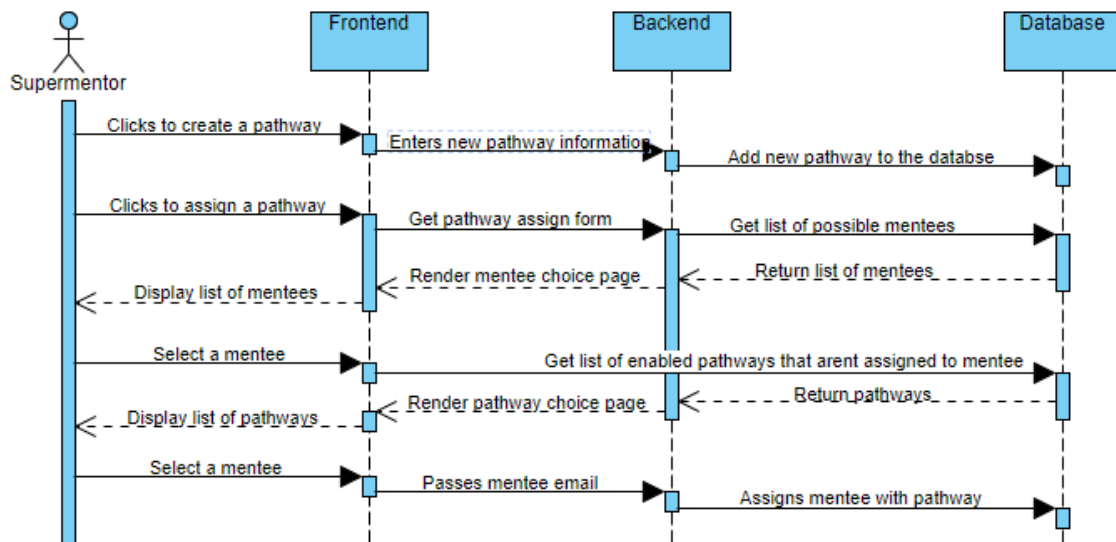
User Logs in & Dashboard is Displayed [Any User]



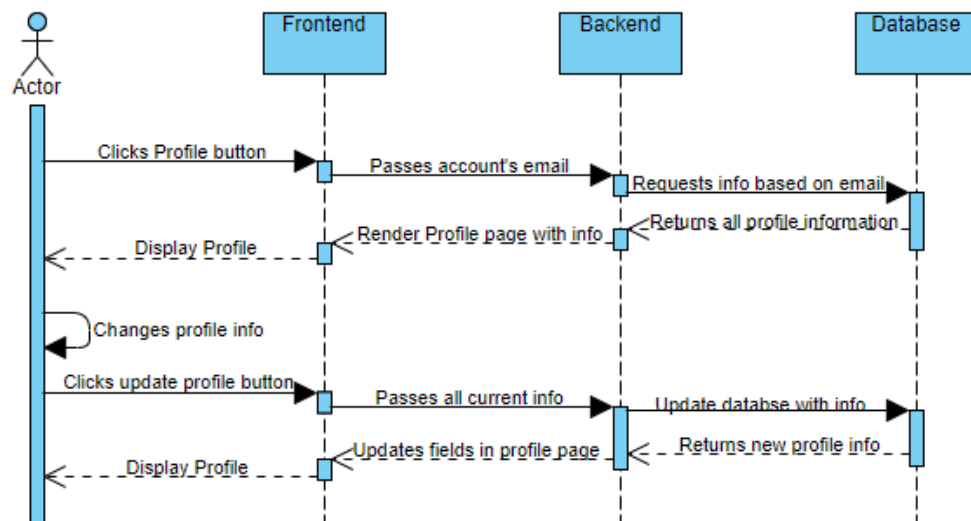
Register New User [Supermentor]



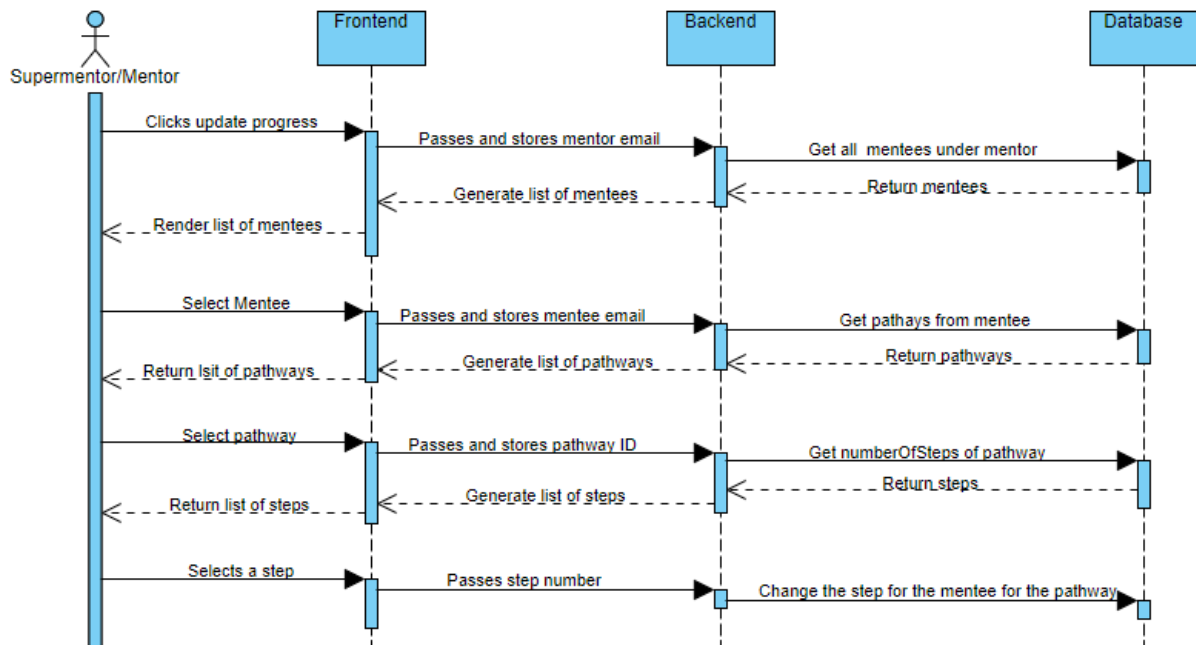
Creating & Assigning a Pathway [Supermentor]



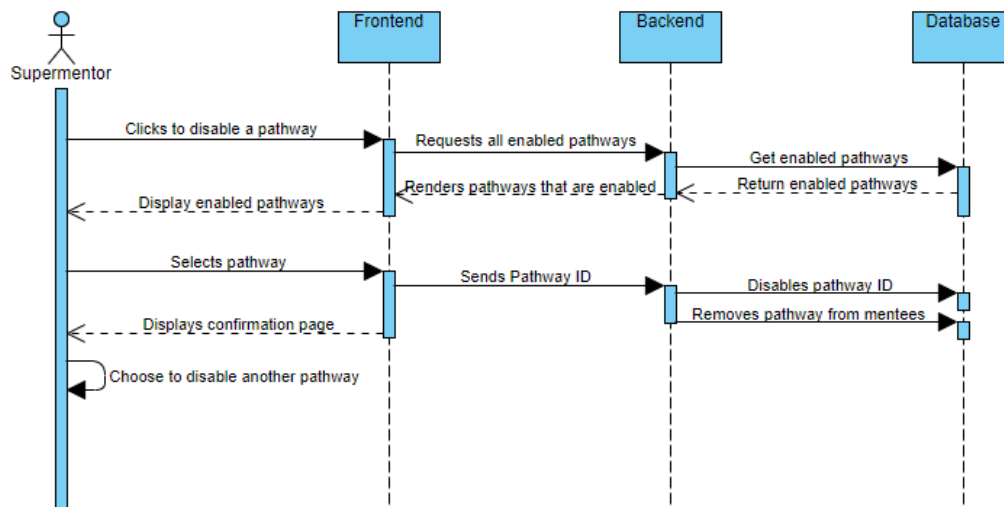
Viewing & Updating Profile [Any User]



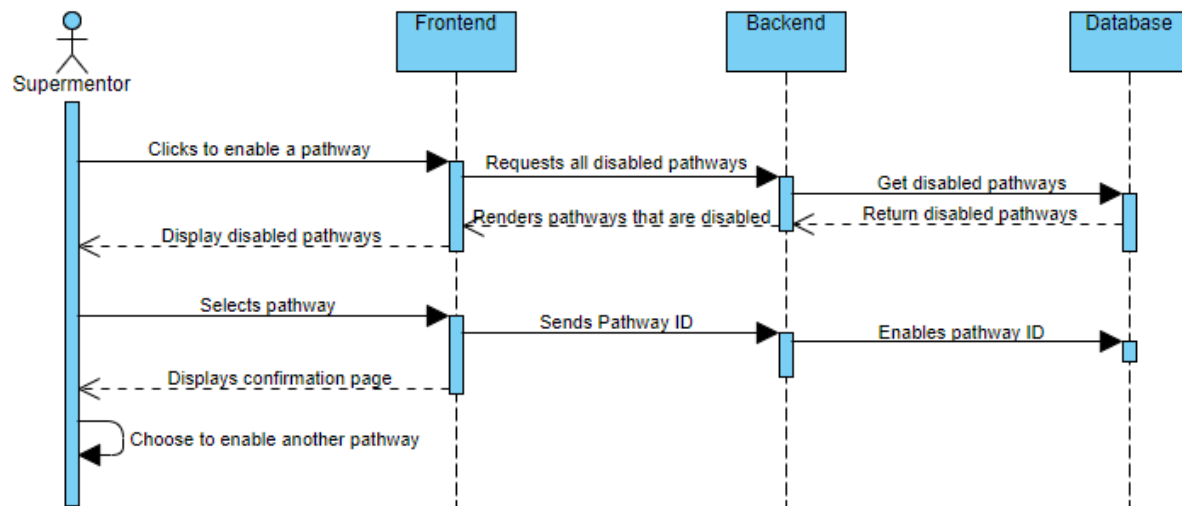
Updating Mentee Progress [Mentor/Supermentor]



Disabling a Pathway [Supermentor]

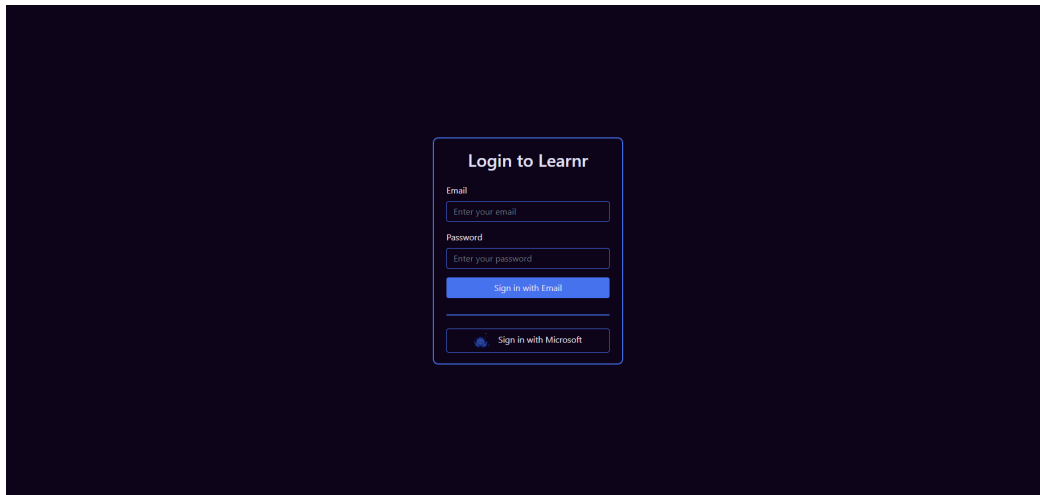


Enable a pathway [Supermentor]

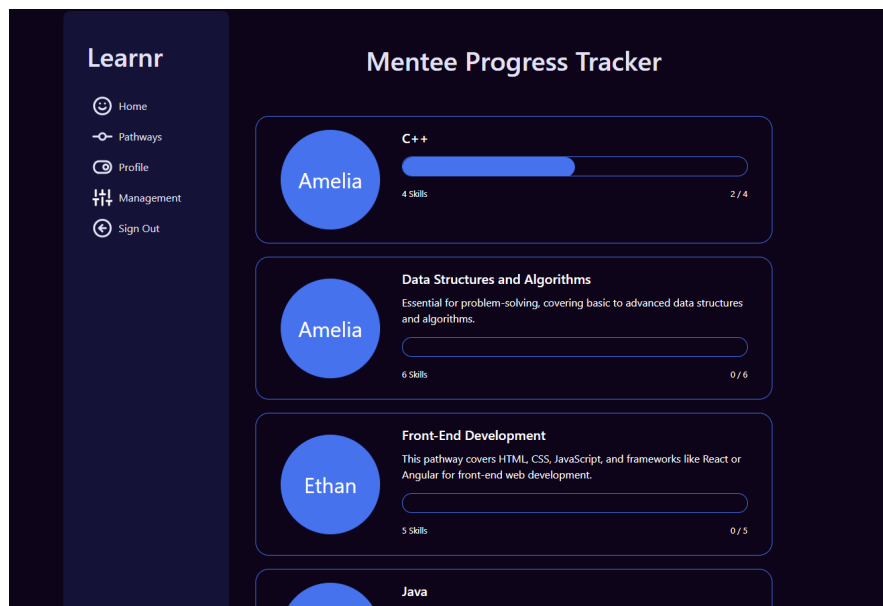


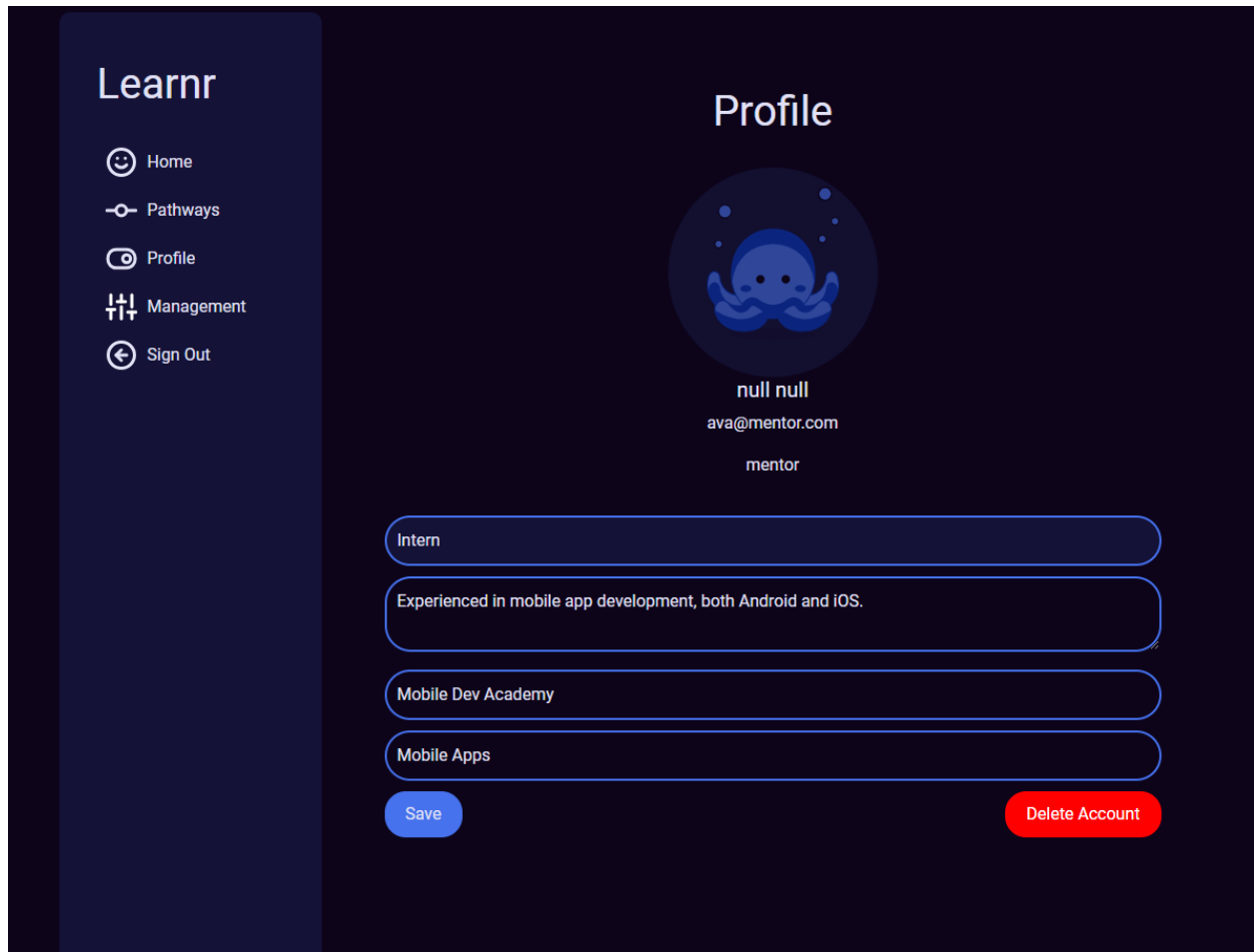
Appendix B: User Interface Design

Login Screen



Mentor/Supermentor Homepage



Profile Page

The screenshot displays the 'Profile' page of the 'Learnr' application. On the left is a dark blue sidebar with the 'Learnr' logo and navigation links: Home (smiley face icon), Pathways (key icon), Profile (eye icon), Management (gears icon), and Sign Out (arrow icon). The main content area has a dark blue background. At the top, the word 'Profile' is centered. Below it is a circular profile picture placeholder containing a blue octopus icon. Underneath the picture, the text 'null null' is displayed, followed by the email 'ava@mentor.com' and the name 'mentor'. Below this information are four rounded rectangular input fields containing the text: 'Intern', 'Experienced in mobile app development, both Android and iOS.', 'Mobile Dev Academy', and 'Mobile Apps'. At the bottom of the main area are two buttons: a blue 'Save' button on the left and a red 'Delete Account' button on the right.

Learnr

- Home
- Pathways
- Profile
- Management
- Sign Out

Profile



null null
ava@mentor.com
mentor

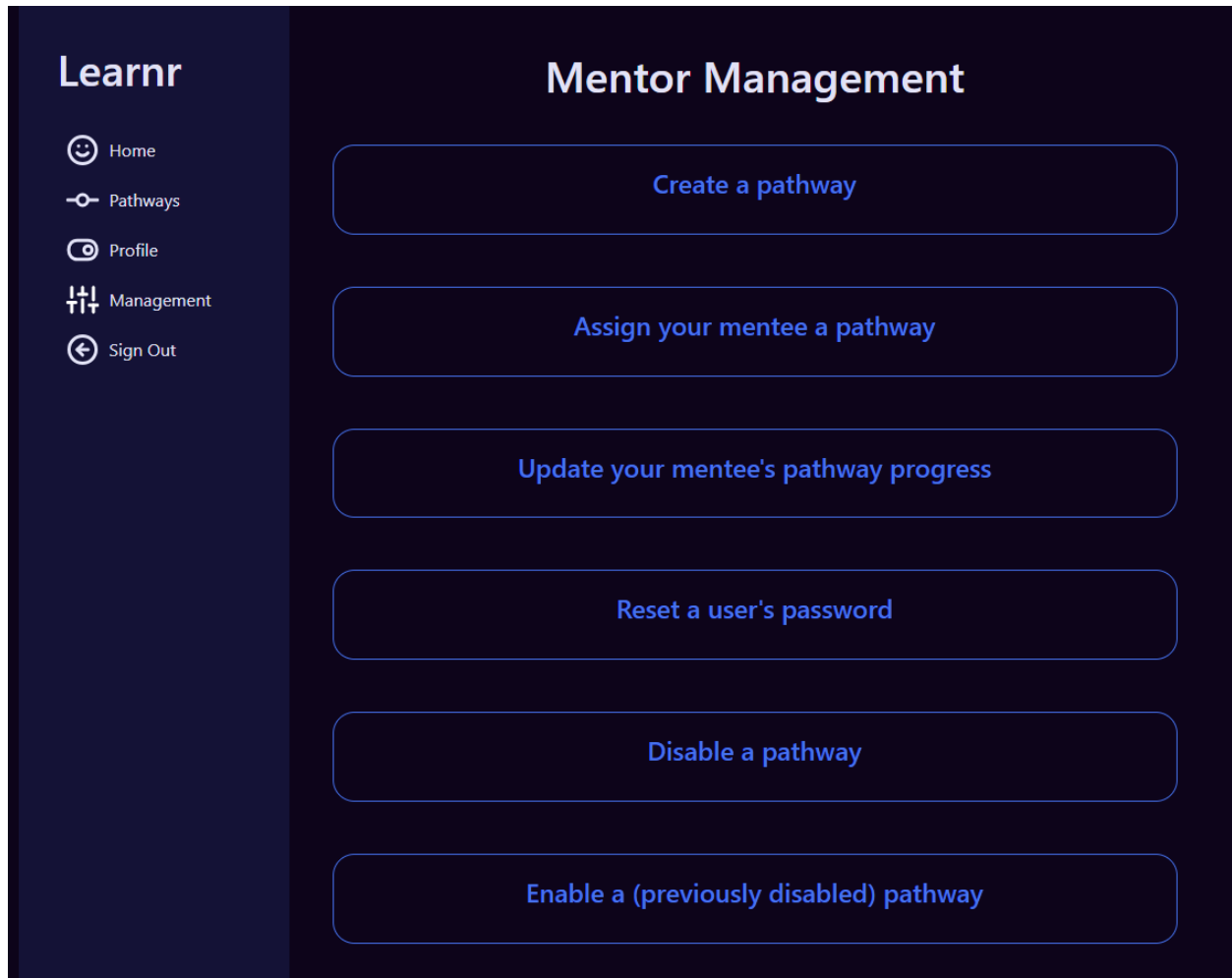
Intern

Experienced in mobile app development, both Android and iOS.

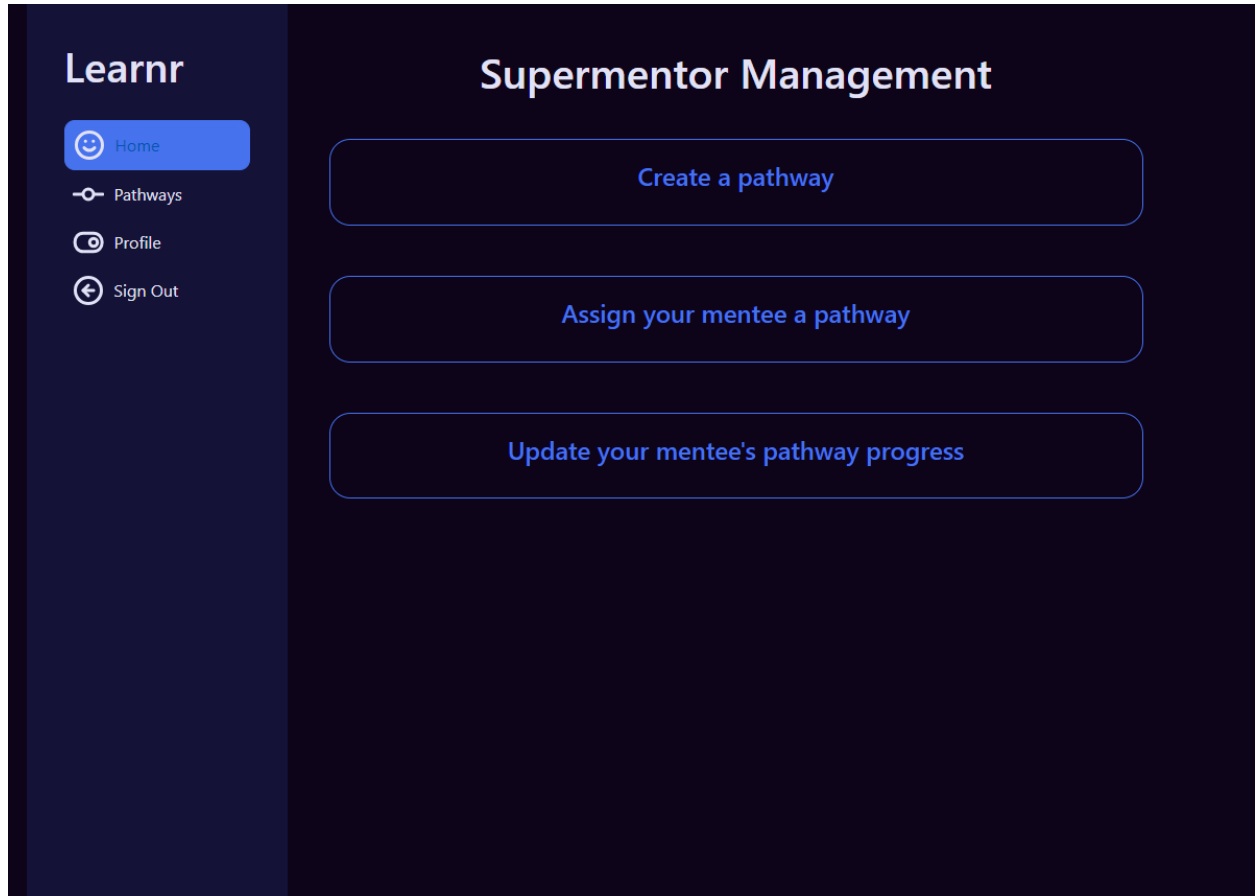
Mobile Dev Academy

Mobile Apps

Save Delete Account

Supermentor Management Page

Mentor Management Page



Appendix C: Sprint Review Reports

Sprint 1

S1 - Review Meeting Minutes

Project: Learnr

Attendees: Rose, Amani, Tyler

Start time: 7:00 pm

End time: 7:30 pm

After a show and tell presentation, the implementation of the following user stories was accepted by the product owners:

- User Story 1: Admin mentor login credentials
- User Story 2: Super user should be able to create mentee account
- User Story 3: Mentee to be able to sign in to account with a code or premade password given by super user with a pre-filled profile

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Sprint Planning meeting.

- As a user, I want the ability to sign in with an email and password
- As a user, I want the website to identify what role I have: users (mentor vs mentee) vs administrator/facilitators
- How this should be reflected in the user story definition in Mingle:
 - we added more definition to the user stories and broke them down further

Sprint 2

S2 - Review Meeting Minutes

Project: Learnr

Attendees: Rose, Amani, Tyler

Start time: 5:00 PM

End time: 5:30 PM

After a show and tell presentation, the implementation of the following user stories was accepted by the product owners: All.

- User Story 1: Admin mentor login credentials
- User Story 3: Mentee to be able to sign in to account with a code or premade password given by super user with a pre-filled profile

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Sprint Planning meeting.

- User Story 2: Super user should be able to create mentee account

Sprint 3**S3 - Review Meeting Minutes**

Project: Learnr

Attendees: Rose, Amani, Tyler

Start time: 6:00 PM

End time: 6:30 PM

After a show and tell presentation, the implementation of the following user stories was accepted by the product owners:

Code restructuring

- User Story 0: As the developer, I want the project directory to have separation of concern by putting all front end files in a client directory, and all server files in a server directory so that it is easier to understand my work space.

Account Management:

- User Story 3: As a super mentor, I want the ability to reset the password of other users, so that I can assist them in case they forget their password or encounter login issues.

Learning Roadmap:

- User Story 5: As a mentor, I want to (skeleton to) assign a learning pathway to a single mentee, so that I can provide personalized guidance and track their progress.

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Sprint Planning meeting.

- User Story 1: As a user, I want to be able to fill up my account profile with additional information to my profile such as bio, interests, school, so that my profile is complete and reflects my identity within the system.

- User Story 2: As a user, I want the ability to edit my account profile information, so that I can keep it up to date as my details change

Sprint 4**Review Meeting Minutes**

Project: Learnr

Attendees: Rose, Amani

Start time: 6:00 PM

End time: 6:30 PM

After a show and tell presentation, the implementation of the following user stories was accepted by the product owners:

1. User Story 1: As a developer, I want to transition the Learnr project from using a CSV database to a relational database, so that we can efficiently manage and access data.
2. User Story 2 **(Revisited)**: As a user, I want to be able to see my profile information - Now using the relational database
3. User Story 3 **(Revisited)**: As a user, I want to be able to fill up my account profile with additional information to my profile such as bio, interests, school, so that my profile is complete and reflects my identity within the system - Now using the relational database. Upon creation of a user, all profile information characteristics are filled in with null values (that show nothing if value is true, and get replaced with actual value if information is edited)
4. User Story 4 **(Revisited)**: As a user, I want the ability to edit my account profile information, so that I can keep it up to date as my details change - Now using the relational database
5. User Story 5 **(Revisited)**: As a super mentor, I want the ability to reset the password of other users, so that I can assist them in case they forget their password or encounter login issues. - Now using the relational database
6. Stretch work:
 1. Revisit the previous User stories and adapt to the new database.
 2. Super mentor can fill in user account with high level info (email, position).
 3. All users have the ability to sign in using a username and password.
 4. User with account created by supermentor can sign in with the username and password created by supermentor.

5. Super mentor can fill in user account with high level info (username, password).
6. Super mentor can create user account (super mentor, mentor, mentee).

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Sprint Planning meeting.

- None

Sprint 5**Review Meeting Minutes**

Project: Learnr

Attendees: Rose, Amani

Start time: 6:00 PM

End time: 6:30 PM

After a show and tell presentation, the implementation of the following user stories was accepted by the product owners:

1. All users have the ability to sign in using a username (only using email) and password.
2. Design of login screen (that preserves functionality).
3. 2 components for the side menu bar (mentor/supermentor and mentee).
4. Mentee single pathway tracker.

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Sprint Planning meeting.

1. Users can set up 2FA on their account.
2. All users have the ability to sign in using their Microsoft Account credentials.

Sprint 6**Review Meeting Minutes****Project:** Learnr**Attendees:** Rose, Amani**Start time:** 6:00 PM**End time:** 6:30 PM

After a show and tell presentation, the implementation of the following user stories was accepted by the product owners:

1. pathway id (what happens when one gets deleted?) - automatic assigned pathway id by hashing the name of the pathway
2. Supermentors can delete learning roadmaps
3. Supermentors and mentors can create a learning road map (form connected to database)
4. skill tracker - mentee
5. assigned mentee dashboard (skill tracker for each mentee) - supermentor/mentor
6. Any user should be able to delete their own account
7. Supermentor and mentors can return to “assign” page after attempting to assign a pathway already assigned to a user

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Sprint Planning meeting.

1. Users should have the option to securely log out of their account.

Sprint 7

Review Meeting Minutes

Project: Learnr

Attendees: Rose, Amani

Start time: 6:00 PM

End time: 6:30 PM

After a show and tell presentation, the implementation of the following user stories was accepted by the product owners:

[Untitled Database](#)

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Sprint Planning meeting.

None

Appendix D User Manuals, Installation/Maintenance Document, Shortcomings/Wishlist Document and other documents

1. User Manuals

Background

Learnr's current implementation can be separated into two sections; the frontend and the backend. The backend is responsible for sending HTTP requests to the frontend as well as handling incoming HTTP requests from the frontend. It is also capable of accessing, creating and updating entries in a MySQL database. The frontend renders all HTML files sent by the backend and can request data from to backend to render pages dynamically.

Screenshots will be provided as needed

Getting Started

To get the application running:

1. Navigate to the project directory in your terminal.
2. Run the following command to start the server:

```
node server.js
```



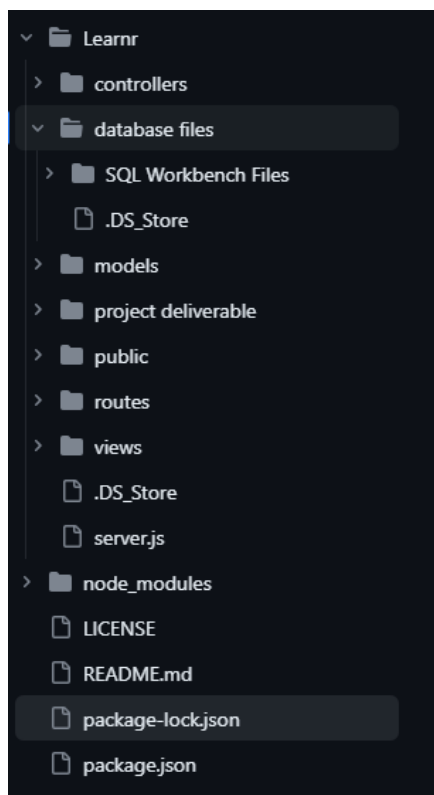
The application will start and run on localhost. Access the system via a web browser.

Usage

1. Access the application via a browser using the URL `http://localhost:3000`.
2. Interact with the application as per your requirements.

File Structure

The structure of the project follows an MVC structure, which stands for Model-View-Controller. It is composed of 3 main folders with a couple of side folders. The model folder holds every file that has to access the database. The model is filled with independent query calls to the database that can be called on in any order or at any time from the controller. The view folder holds every html and ejs file that will be rendered by the frontend. The controller folder holds a file that allows for the view to communicate with the model. For example, when registering a new user, the view will display the registrarNewUserForm.html file and send any input information to the controller. The controller will take in the information and pass it as variables to the correct model file that can handle adding a new user to the database. The public folder holds static js file that arent meant to be manipulated, while the router folder contains redirect paths for the frontend.



2. *Installation/Maintenance Document*

Follow the steps below to set up the project on your local machine:

1. Node.js Installation:

- Download and install Node.js version (v18.17.1) from [Node.js official website](#).
- Use the recommended installation settings during the setup.

2. Project Setup:

- Clone the repository to your local machine.
- Navigate to the project directory in your terminal.

3. Dependency Installation:

- If the following modules are not installed, you can download and install them by running the given commands:

```
npm install express
npm install bcrypt
npm install csv-parser
npm install mysql2
npm install @azure/msal-node
npm install ejs
```



4. MySQL Setup:

- Download MySQL from this website: <https://dev.mysql.com/downloads/installer/>
 - For windows select the download with the higher download count.
 - For mac select the option that works for your specified operating system.
- Once downloaded use all default settings and when it asks to enter a password type the recommended: 'root' or 'password' (This password may vary from computer to computer).
- If possible do a full installation which will include mysql workbench and mysql sever.
 - Run mySQL workbench and select the database and log in using the password you set for yourself.

3. *Wishlist*

The following lists are features that we believe should be implemented in order importance and by category:

☐ Account Authentication and Security

Provide users with a secure option to log out of their account	Account Management	Must	XS	Not started
Add Ability to Amend Mentee Progress from Mentor/Supermentor Dashboard - frontend	Account Management	Should	M	Not started
Identify the additional abilities required by mentors	Account Management	Should	M	Not started
Enable all users to sign in using their Microsoft Credentials	Account Management	Should	L	Not started
Identify the additional abilities required by admins	Account Management	Should	M	Not started
Add ability to ammend user name on profile - backend	Account Management	Should	XS	Not started
Enable users to set up 2FA on their account	Account Management	Could	M	Not started
Identify the additional abilities required by mentees	Account Management	Should	M	Not started
Add Ability to Amend Mentee Progress from Mentor/Supermentor Dashboard - backend	Account Management	Should	M	Not started
Allow administrators to delete user accounts	Account Management	Could	XS	Not started
Allow supermentor to register another user after completing one registration	Account Management	Must	XS	Not started
Enhance Logout Mechanism for Secure User Sessions	Account Management	Must	M	Not started
Enable administrators to designate mentors and mentees as coaches for skill sharing	Account Management	Could	L	Not started

☐ Learning Roadmaps

Aa Name	≡ Features	⌵ MoSCoW	⌵ Size	⌵ Status
Enable super mentors and mentors to assign multiple learning roadmaps to a single mentee	Learning Roadmap	Could	M	Not started
Allow supermentors and mentors to enhance learning roadmaps with external resources (links, documents, videos)	Learning Roadmap	Could	L	Not started
Enable super mentors and mentors to create and save learning roadmaps for public or private use, detailing skills and acceptance criteria	Learning Roadmap	Could	XS	Not started
Enable super mentors and mentors to create and save a basic new learning roadmap outlining skills and acceptance criteria	Learning Roadmap	Could	M	Not started
Enable super mentors and mentors to assign a chosen learning roadmap to multiple mentees	Learning Roadmap	Should	M	Not started
Enable super mentors and mentors to assign multiple learning roadmaps to multiple mentees	Learning Roadmap	Could	M	Not started

☐ Rewards System

Display mentees' badges and trophies on their account profile and dashboard	Trophies and Rewards	Should	M	Not started
Empower mentees to share their badges and trophies on social media platforms like LinkedIn, Twitter, etc.	Trophies and Rewards	Should	L	Not started
Allow mentees to earn badges and trophies as they progress in learning pathways	Trophies and Rewards	Should	M	Not started
Provide option for mentees' badges and trophies to be publicly or privately visible on their profile and dashboard	Trophies and Rewards	Could	M	Not started

REFERENCES

- Node.js official website <https://nodejs.org/en>
- MySQL Download. <https://dev.mysql.com/downloads/installer/>
- Agile Methodology. <https://www.agilealliance.org/agile101/>
- Bootstrap Framework. <https://getbootstrap.com/>
- Express.js Documentation. <https://expressjs.com/>
- Figma Tool. <https://www.figma.com/>
- Git Version Control. <https://git-scm.com/>
- GitHub. <https://github.com/>
- MVC Architecture. https://www.tutorialspoint.com/mvc_framework/mvc_framework_introduction.htm
- Node.js. <https://nodejs.org/>
- Node Package Manager (NPM). <https://www.npmjs.com/>
- Relational Database Concept. <https://www.oracle.com/database/what-is-a-relational-database/>
- SQL Language. <https://www.iso.org/standard/63555.html>
- StarUML Tool. <http://staruml.io/>
- UML Diagrams. <https://www.uml-diagrams.org/>
- Visual Studio Code. <https://code.visualstudio.com/>