

Database Project

Shin Hyung Lee
Joseph Chang
Dung Dong
Truong Phu Vu
Vy Dinh
Khang Huynh

December 8, 2022
CS 160
Group 4: SQL4U
(Final) Milestone 3

Table of Contents

Who did what	4-7
Website and Password Information	8
Website Location	8
Password Information	8
Tasks Accomplished Checklist	8-9
Implementation of core features	10-28
At least 4 of your proposed features implemented	10-19
Emerging technology used for your project	20-22
Lack of expertise that could have helped you in this project	23-24
New skill learned and applied to this project	25-26
Need for continuous professional development	27-28
Website implementation and public web hosting preparation process	28-32
Show its main components and how they rendered your dynamic website (illustrate with a diagram)	33-43
Taxonomy	33-43
Website Testing	44-49
Selenium	44-48
Testing Platform Information	48-49
For each of your implemented proposed features:	49-81
A typical flow of your website with narrated snapshots	49-64
Provide relevant final spec, use cases, and use case diagrams.	65-81
Version control system to collaborate	82-89
User Guide	82-89
One complete process on Github (or similar)	82-89
Additional things (not mentioned above)	89
Design document	90

Site Map Flowchart	93
Describe agile design process	98

Who did what

What	Description	Who
Figma's prototype creation	Created website prototype by using Figma	Shin Hyung Lee Joseph Chang Dung Dong Truong Phu Vu Vy Dinh Khang Huynh
webpage creation for (Login/Register) feature	Created Login and Register page Added constraints for input fields Style the page to look similar to the pages' prototypes.	Dung Dong Truong Phu Vu
webpage creation for (User Profile) feature	Created User Profile Page	Joseph Chang, Dung Dong
webpage creation for (Learning Hub) feature	Learning Hub is the first feature where users can learn about the technical subject (database) Implement courses catalog, where user can access all courses covering technical subject If user want to see a list of courses which have the same topic, they can click their favorite topic Each course has a list of tutorial. After users finish the list, they we'll be able to see relative projects Style the Learning Hub page Adding functional buttons	Khang Huynh Dung Dong

webpage creation for (Interview Questions) feature	<p>On the Interview Questions page, learners can practice interview questions. Upon clicking on the “answer” button, they can view answers. If they like particular questions, they can click on “save question” button to save the question to their “saved questions” list.</p>	Shinhyung Lee Dung Dong
webpage creation for (Q&A) feature	<p>QnA is like a discussion forum, which is the fourth feature. Users can create and edit their own questions, or add answers. Style the QnA page Add functional buttons and features</p>	Vy Dinh Dung Dong
webpage creation for (Recommended Project) feature	Recommended Project feature is one of the chosen features which provides learners with related projects for each topic.	Truong Phu Vu
Website's styling generalization	Set up website general appearance using pure CSS.	Truong Phu Vu
Website's styling finalization	Added and fixed each feature's styling. Synchronized each feature's styling. Obtained website's responsiveness on 3 types of screens: desktops, tablets, and mobile.	Truong Phu Vu

Work on hosting webpages in remote server	Hosted both backend/frontend servers on AWS	Dung Dong
Created Backend Server / API	Provided API endpoints for all features.	Dung Dong
Services for the frontend server to interact with the database.	Created services to send HTTP requests to the backend server for all features. Connected backend server/database to all features.	Dung Dong
Validating revised spec on the assigned team	Revised specs according to the validation doc	Shin Hyung Lee Joseph Chang Dung Dong Truong Phu Vu Vy Dinh Khang Huynh
Revised Sitemap Flowchart	Revised the sitemap flowchart due to changes to the website flow since last milestone	Joseph Chang
Revised overview use case diagram	Revised the overview use case diagram due to changes to the since last milestone	Joseph Chang
Powerpoint Preparation	Create and add content for the presentation	Shin Hyung Lee Joseph Chang Dung Dong Truong Phu Vu Vy Dinh Khang Huynh
Report	Create, edit, and finalize report for the final milestone	Shin Hyung Lee Joseph Chang Dung Dong Truong Phu Vu Vy Dinh Khang Huynh

Powerpoint Presentation	Powerpoint presentation to describe the final milestone achievement	Shin Hyung Lee Dung Dong Truong Phu Vu Vy Dinh Khang Huynh
Agile Design	Collaborated and communicated throughout the entire project. Update each members on individual progress, challenges, and seek assistance if necessary	Shin Hyung Lee Joseph Chang Dung Dong Truong Phu Vu Vy Dinh Khang Huynh
Testing	Using Selenium IDE to create 9 test cases for user and admin Record one test suite on one complete workflow related to our application Test website against various popular browsers and browser version	Khang Huynh Vy Dinh

Website and Password Information

Website Location

<https://www.sql4u.dev/>

Backend Server/ API: <https://api.sql4u.dev>

Password Information

Admin: [sql4u@gmail.com] [SQL4U!@]

Non-admin account 1: [cs160-1@gmail.com] [password1]

Non-admin account 2: [cs160-2@gmail.com] [password2]

MongoDB Atlas Cloud Database user account

Database: [SQL4U] [GYHpCzypi18yGqKG]

Tasks Accomplished Checklist

Copy and paste the checklist from the final project description and replace the table content below. Put “X” only next to items completely implemented.

Double penalty for claiming items not worked on.

X	A. Implementation of core features
X	A.1 at least 4 of your proposed features implemented
X	A.2 Emerging technology used for your project
X	A.3 Lack of expertise that could have helped you in this project

X	A.4 New skill learned and applied to this project
X	A.5 Need for continuous professional development?
X	B. Server-side script hosting
X	B.1 Provide one admin account and 1 non-administrator accounts
X	B.2 Upload your web script to your public website
X	B.3 Share the public web link on the report
X	C. Software requirement, development
X	C.1 A design document
X	C.2 Validated and modified use cases for each implemented feature
X	C.3 Use case diagrams
X	C.4 Agile design description
X	D. Software versioning
X	D.1 Show snapshots of one complete check-in and check-out version control process.
X	D.2 Add the instructor and the TA to your Github repository (or similar).
X	E. Testing
X	E.1 Use Selenium IDE on Firefox or similar to record one test suite on one complete workflow (including a login) in your website
X	E.2: Summarize in a table the testing against various popular browsers and versions.
X	F. Compose technical reports and participate in technical discussion
X	F.1: Provide a technical report following the report template
X	F.2: Participate and collaborate on Github and Discord
X	G. Peer evaluation
X	a. Upload confidential evaluation
X	b. Provide web-based peer evaluation.

Implementation of core features

At least 4 of your proposed features implemented

A.1 Present a working website that shows at least 4 of your proposed features. The content and hierarchy need to be consistent with the features and the technical area you proposed. Each feature should have at least 3 child pages with relevant content in words and images

1. Document a typical flow of your website with snapshots of webpages along with comments explaining what is what.
 2. Offer at least one page of narrated description with screenshots for each feature implemented
-
1. User Profile Dashboard
 - a. <https://www.sql4u.dev/profile>
 - b. User profile where you can see personalized content. The user can see the courses, interview questions they've saved, finished projects and can go directly to ask a question. Since all the pages that can be accessed from the user profile dashboard are individual features so all the snapshots will be redundant, we opted not to post anymore snapshots here.

The screenshot shows a user profile dashboard with the following sections:

- Courses you might interested**:
 - Database I: Intro to Databases
 - Test98
- INTERVIEW QUESTIONS**:
 - Questions you've saved
 - What are the differences between a DBMS and RDBMS?
- FINISHED PROJECTS**:
 - Completed Projects
 - Project #3 - HASH INDEX
 - Project #4 - QUERY EXECUTION
 - Project #2 - Telephone
- QUESTIONS**:
 - Any Questions?
You can post/find questions here:
Ask!

2. Learning Hub
- a. <https://www.sql4u.dev/catalog>
3. Learning hub acts as a homepage where the users land after signing in. It is a hub where users can see learning contents related to topics. Users can use the filter on top to select a specific topic or simply browse them all.

The screenshot shows a 'TOPIC' section with a 'SQL' filter selected. Below are six course cards:

- Database I: Intro to Databases** by Adam Wilbert (5.0 stars, 3h 21m) - Go!
- Database II: Data Structures** by Adam Wilbert (5.0 stars, 2h 55m) - Go!
- Database III: Database Management** by Adam Wilbert (5.0 stars, 2h 14m) - Go!
- Learning SQL Programming** by Scott Simpson (5.0 stars, 1h 20m) - Go!
- Advanced SQL for Data Scientists** by Dan Sullivan (5.0 stars, 2h 30m) - Go!
- SQL vs. NoSQL: Which Type Is Right for You?** by David Jones-Gilardi (5.0 stars, 0h 29m) - Go!

Page navigation: 1 2

- a. Users can click into the topic and it will lead them to course catalog where the users can see all the learning contents that pertains to the topic.

[Go back](#)

Database I: Intro to Databases

1. Introduction
2. Database Core Concepts
3. How to Set Up a Database Playground?
4. Graphical Interfaces
5. Create Table Objects
6. Add Data to a Table
7. Retrieve Information
8. Manipulate Data
9. References
10. Projects

We Picked Some Projects For You To Practice Your Skills

[Let's Go >>](#)

- b. When the users click into the list of learning content, they'll be directed to the actual content where they can read on the topic.

Database Core Concepts

What is a database?

--> A database is an organized collection of material that has been arranged and is often kept electronically in a computer system. A database management system often oversees a database (DBMS). The term "database system," which is frequently abbreviated to "database," refers to the combination of the data, the DBMS, and the applications that are connected to it.

--> To facilitate processing and data querying, the most popular types of databases currently in use typically model their data as rows and columns in a set of tables. The data may then be handled, updated, regulated, and structured with ease. For writing and querying data, most databases employ structured query language (SQL).

Relational database management systems

--> A common type of database that stores data in tables so that it can be used in relation to other stored datasets is a relational database management system (RDBMS or simply RDB). As opposed to flat files or hierarchical databases, relational databases are the most common type of database used by businesses today.

The client-server model

1. Database Server:
 - > Manages all relational database responsibilities
 - > Organizes, stores, and protects your data.

4. Recommended Project

- a. <https://www.sql4u.dev/projectLandingPage>
- b. Recommended projects can be accessed through the learning hub or directly from the menu on top. We integrated recommended projects to be a part of the learning hub to offer more seamless learning.

The screenshot shows a learning hub interface with a light blue header. At the top, there is a navigation bar with tabs: All, Basic Concepts, Relational Model, File Organization, NoSQL, and SQL (which is highlighted). Below the header, there are three project cards, each featuring a circular icon with a gear and monitor, a title, a rating, an author photo, and a duration. Each card has a 'See Projects' button at the bottom right.

Project Title	Rating	Author	Duration	Action
Database I: Intro to Databases	5.0	Adam Wilbert	3h 21m	See Projects
Database II: Data Structures	5.0	Adam Wilbert	2h 55m	See Projects
Database III: Data Management	5.0	Adam Wilbert	2h 14m	See Projects

- c. Users can click into the topic and see projects related to each topic. Once the user is on the project list page, they can choose which project they want to look at.

Database I: Intro to Databases

Project #1 - SQL - BEGINNER

Project #2 BUFFER POOL - INTERMEDIATE

Project #3 - HASH INDEX - PROFESSIONAL

Project #4 - QUERY EXECUTION - PROFESSIONAL

Project #5 - BEGINNER

[Go Back](#)

- d. When the user chooses a specific project and clicks into it, the user will be led to the project description page. The feature also provides a check mark once the user has completed the projects.

Project #1 - Palindrome Checker

INSTRUCTION

Return true if the given string is a palindrome. Otherwise, return false.

A palindrome is a word or sentence that's spelled the same way both forward and backward, ignoring punctuation and spacing.

Note: You'll need to remove all non-alphanumeric characters (punctuation, spaces and symbols) and transform the string to the same case (lower or upper case) in order to check for palindromes.

We'll pass strings with varying formats, such as racecar, RaceCar, and race CAR among others.

We'll also pass strings with special symbols, such as 2A3*3a2, 2A3 3a2, and 2_A3*3#A2.

TESTING

palindrome("eye") should return a boolean.

palindrome("eye") should return true.

palindrome("_eye") should return true.

palindrome("race car") should return true.

palindrome("not a palindrome") should return false.

palindrome("A man, a plan, a canal. Panama") should return true.

palindrome("never odd or even") should return true.

palindrome("nope") should return false.

palindrome("almostomla") should return false.

palindrome("My age is 0, 0 si ega ym.") should return true.

palindrome("1 eye for of 1 eye.") should return false.

palindrome("0_0 (: /\\ :) 0-0") should return true.

palindrome("five|_|four") should return false.

 Finished

Go back

Go to Top

5. Interview Question

- a. <https://www.sql4u.dev/interview>
- b. Interview questions can be accessed through the top navigation bar. The user can see a list of interview questions. The page also has a filter system by topic which enables the user to narrow down the interview questions they are interested in.

The screenshot shows a mobile application interface for interviewing. On the left, there's a sidebar titled 'TOPICS' with a list of database concepts: Basic Concepts, Relational Model, File Organization, NoSQL, and SQL. Below this is a 'Saved Questions' section with a heart icon. At the top right is a large header 'Interview Questions'. Below it are three cards, each containing a question, a 'Save Question' button, and an 'Answer' button with a plus sign.

- Why would we need to use NoSQL??
Save Question Answer +
- Briefly explain what horizontal scaling is?
Save Question Answer +
- What is vertical scaling?
Save Question Answer +

- c. The user can click the answer button to reveal the answer to the question.

The screenshot shows a detailed view of the first interview question. The question is 'Why would we need to use NoSQL??'. Below it is a large yellow box containing the answer: 'At the present time, the internet is loaded with big data, big users, big complexity, etc. It is also becoming more complex day by day. NoSQL is answer to all these problems; It is not a traditional database management system, a relational database management system (RDBMS). NoSQL can handle and sort all type of unstructured, messy and complicated data.' There is a 'Save Question' button at the bottom.

Why would we need to use NoSQL??
Save Question Answer -

At the present time, the internet is loaded with big data, big users, big complexity, etc. It is also becoming more complex day by day. NoSQL is answer to all these problems; It is not a traditional database management system, a relational database management system (RDBMS). NoSQL can handle and sort all type of unstructured, messy and complicated data.

Briefly explain what horizontal scaling is?
Save Question Answer +

- d. Users can also choose to save any interview questions to create their own list. The

curated saved interview questions can be viewed by clicking into saved questions button.

- e. User can also unsave question from “Saved Questions” list by clicking on Unsave Question button

The screenshot shows a mobile application interface. On the left, there's a sidebar titled "TOPICS" containing links to "Basic Concepts", "Relational Model", "File Organization", "NoSQL", and "SQL". Below these is a "Saved Questions" section with a heart icon. On the right, the main content area has a header "Interview Questions". It displays three interview questions with "Unsave Question" buttons and "Answer +" buttons:

- Briefly explain what horizontal scaling is?
- What are the differences between a DBMS and RDBMS?
- Why would we need to use NoSQL??

6. Q&A

- a. <https://www.sql4u.dev/qna>
- b. Q&A page is where the users can go and ask questions related to the database. This page can be accessed from the top navigation bar and user profile dashboard. Users can type in the question, choose the topic, type in the description of the question and click on the post button.

The screenshot shows a user interface for a Q&A section. On the left, there's a sidebar with a 'Q&A' icon and a list of topics: Basic Concepts, Relational Model, File Organization, NoSQL, and SQL. The 'SQL' topic is highlighted. On the right, there's a main area where a user named 'UserName' has posted a question: 'Is Database a one word or two words?'. The question is categorized under 'Basic Concepts'. A blue 'Post' button is visible at the bottom of the question box. A small downward arrow is located at the bottom right of the main content area.

- c. Once the user clicks on the post button, the question will be listed on the bottom part of the page along with other questions already posted by other users.

Basic Concepts

Relational Model

File Organization

NoSQL

SQL

Description

Post

▼

sql4u@gmail.com
2022-12-06T05:46:31.718Z
Topic: Relational Model
Hello world

cs160-1@gmail.com
2022-12-08T07:20:48.741Z
Topic: Basic Concepts
Is Database a one word or two words?

- d. Users can click into individual questions to view them and reply to that specific question by typing in the reply and clicking the reply button. Once the user clicks on the replies to the question, the reply will show on that question page.



cs160-1@gmail.com



Topic: Basic Concepts

Title: Is Database a one word or two words?

Description:

I've seen people write it as database or data base? It is supposed to be one word or two words? I'm serious



Reply

Reply

Delete



cs160-1@gmail.com

2022-12-08T07:20:48.741Z

It's one word obviously, google it

Emerging technology used for your project

(one paragraph per team member)

A.2 Describe a technology that you used in this implemented website project.

- Vy Dinh:

In this implemented website project, my job is focused on frontend of a QnA page feature. First, I used Figma to design the UI for the page with desired features, style, and color. Next, I started to design the skeleton of the page with HTML and CSS. I also learned and used Bootstrap for this. Our team uses React.js, so after coding with HTML and CSS, I am integrating it into React.js since there are some differences between HTML and React.js. To collaborate with other team members' work, I have a chance to work more on git and GitHub. I encounter some errors while using git, which gives me some chances to learn more about git commands. Moreover, I also know how to create a JSON files for the database for this feature. Finally, to complete the project I used Selenium to test the webpage functions and flow.

- Dung Dong:

- Backend Server: Express.js, MongoDB, mongoose, jsonwebtoken, bcrypt, joi, lodash:
 - Express.js: to create a RESTful API server and register endpoints for the server to interact with the database.
 - mongoose: to establish a connection between the backend server and the MongoDB database and create a database model so that the server can interact with the database through it.
 - jsonwebtoken: to generate jwt and decode it.
 - bcrypt: to encrypt a password before storing it in the database.
 - joi: to validate the body of the request sent to the server.
 - lodash: allow to perform a lot of functionality of arrays easily.
- Frontend Server: React.js, react-router-dom v6, Bootstrap, create-react-app, Axios.
 - create-react-app: set up and initialize a default React project.
 - React.js: render the website dynamically with given modules in React.
 - react-router-dom v6: to handle routing in the frontend server as well as provide a way to pass/retrieve information/data through pages.
 - Bootstrap 5.2: simply style a page without writing any CSS.

- Axios: an HTTP service that handles sending requests to the backend server / RESTful API.
- Hosting: AWS Amplify, Elastic Beanstalk AWS:
 - Amplify: host frontend server with the public URL.
 - Elastic Beanstalk: Host the backend server with the public URL so that the frontend server can send requests to the server.
- CI/CD: Git, GitHub: to synchronize the works of all group members together.
- Truong Phu Vu:
 - In this implemented website project's frontend, I use ReactJS as our main technology to implement. However, before going into coding the website interface, I got learn how to use Figma to effectively prototype the UI/UX of the website in order to provide the team with a better vision of how the website will look and function. In terms of implementation, I got to experience more ReacJS to code the UI. In order to complete the first Tier, which is the Presentation tier, HTML, CSS, and Bootstrap also played an important part in creating the structure (HTML) and the styling of the page (CSS, Bootstrap). Besides, navigation between pages is a must for every website. Thus, we utilized Reat-router-dom, an npm package, to accomplish this task. Specifically, we used BrowserRouter, Routes, Route, Link, useParams, useLocation, and Navigate components. Debugging is one of the most crucial parts of coding, and the front end is not an exception. We utilized the browser's Developer Tool and React Developer Tool for debugging our codes. The Presentation tier communicates with the other tiers through the application program interface (API). Thus, Axios HTTP middleware service takes a huge role in this situation.
- Shin Hyung Lee:
 - For this project, I took part in programming the UI(user interface) of the Interview page for our website.
 - For the skeleton part, we as a team used figma to design our barebone. I watched some youtube tutorials to learn the basic figma. I designed the UI for the Recommended Projects page.
 - For the actual programming part, I designed the UI of Interview Questions page with React and Bootstrap. Code sharing/version control was done through git/github. I wasn't entirely familiar with these languages/frameworks. So I learned these from LinkedIn Learning and some other tutorial pages.
- Khang Huynh:
 1. In this project, I focused on designing and front-end development for Learning Hub

- a. Designing: I took some tutorials where I can learn and find the idea to create this feature. Using Figma, I designed the UI for Learning Hub.
 - b. Front-end Server: React.js, react-router-dom v6, create-react-app
 - i. create-react-app: establish and launch a default React project.
 - ii. React.js: using the provided React modules, dynamically render the website.
 - iii. react-router-dom v6: manage routing in the frontend server and offer a means of passing
- 2. Also, I used Selenium IDE on Chrome and Firefox to record one test suite
 - a. Selenium IDE: using this Javascript framework, I take recordings of how our website is used, play them back, and add assertions.
- Joseph Chang:
 - We started off by collaborating with Figma to map out the UI of individual web pages. I developed the user profile page starting with HTML5 and CSS3. Learned how to use React JS since that was the choice of the library chosen by the team. Been a while since I used Bootstrap and Font-Awesome so I needed a refresher. The most challenging part was learning how to use React Components along with HTML, CSS, Bootstrap and Font-Awesome. I used these technologies before but it wasn't with React components. Also, learned how to use GitHub to collaborate on the project

Lack of expertise that could have helped you in this project

(one paragraph per team member)

A.3 Describe one area of lack of expertise that could have helped you in this project. Provide a list to describe this for all members of your team.

- Vy Dinh:

- React.js and git are new to me, so it takes me time to learn about it. Even when working with ReactJS, there are some different ways to write code with class or function. This confused me a bit when I started learning it. I also have some problems when using merge in git. The conflict and some files could affect the other works, but I could fix that with the help of the team. I think I could finish the page earlier with the knowledge of React.js and git.

- Dung Dong:

- Backend server / API automated testing.
 - It appeared to have a lot of errors when I coded the backend server due to not having error handlings back then. The server crashed several times.
 - I tested the endpoints manually, so I could not cover all cases. Only debugged when team members encountered errors related to the server and informed me about that.
 - It would be more time-saving if I were able to implement automated testing for all endpoints.

- Truong Phu Vu:

- I am mainly responsible for the general Front End implementation of the project, therefore, having known of a JavaScript testing utility for React such as Enzyme.JS would have been immensely helpful for me to keep track of the functionality of each component. Moreover, it will save me time on finding which component is causing errors leading to unsuccessfully inoperable features. In terms of coding the features, I wish I had known more techniques to style the pages using CSS and Javascript to make them more responsive and complete.

- Shin Hyung Lee:

- Figma: It was my first time learning figma. Thankfully, there were some good tutorials out there. One youtube video walked through the most essential figma components, and I learned a lot from the video.
- React: I was not familiar with React. I learned some basics from LinkedIn Learning, and other tutorials. In the past, I dabbled with this framework a bit, but surprisingly I couldn't remember anything. I realized it is very important to learn any programming language and program something with it. Otherwise, things I learned, I would not remember.

- Bootstrap: I knew a bit of CSS, but Bootstrap was new for me as well. They have a nice official documentation, so it helped my learning.
 - Github: we had a session of github tutorial during lecture, but it wasn't enough. Other seasoned teammates helped me when I faced problems with github.
- Khang Huynh:
 - Figma: This is the first time I used this tool to create the UI. I spent some hours learning how to use it and finding the design idea.
 - ReactJS: I did not have much experience with it. Before starting the project, I reviewed some material on LinkedIn Learning. During the implementation of my feature, I asked my teammates for help.
 - Git and GitHub: Using merge in git presents some issues for me as well. Other works may be impacted by the disagreement and some files, but I could resolve that with the team's assistance.
 - Selenium IDE: I never use this tool before. To solve this problem, I spent time following the Selenium IDE tutorial and the professor's lectures.
- Joseph Chang:
 - Definitely knowledge with React JS would've helped save time and development cycle. The time spent on learning React JS and how to properly use their components was a tough challenge especially given the short amount of time we needed to complete the milestone. Also, knowledge of GitHub would've been helpful since it was another area that I was completely lost in and if it wasn't for my team, I probably would've never figured it out.

New skill learned and applied to this project

(one paragraph per team member)

A.4 What new skill have you learned and apply to solve what problem in this project?

- Vy Dinh:
 - The first skill I learned is Figma, which was so helpful in building a consistent feature in the whole website with other team members. This made it easier for the team to express our ideas. I also learned how to reverse the code to eliminate the problem in the newly written code by using – git reset. This is the first time I used Git to collaborate with others, like creating forks, syncing work, and solving conflicts after pulling. I think this is the most helpful for us to work in a team. Lastly, working with the team in discord following scrum methodology helps me to arrange time efficiently and suitably to complete work for each milestone.
- Dung Dong:
 - CI/CD as Git and GitHub
 - Know how to sync the works of all members together.
 - Resolve conflicts.
 - Handle pull requests.
 - SCRUM and AGILE:
 - Validating other teams' works and listening to their validation is the best way to practice the AGILE process.
 - We also practiced the SCRUM methodology in Discord, in which each team member does a quick stand-up meeting through chat, summarizing what we have done, what we are about to do, and challenges. This helped us to have an overlook of all team members.
- Truong Phu Vu:
 - After finishing implementing our projects, I got to learn more about:
 - Interpersonal skills
 - My technical writing skills have improve significantly through every milestones report, I have done some research on terminologies which are commonly used in software engineer to fully transfer what I implemented and done into professional sentences.
 - I also learned how to communicate with my teammates in order to remain professional when it came to protecting my ideas.

- Collaboration in project/coding:
 - I got to practice Agile methodology in each milestone through the other teams in the class evaluate my team's work. I also got to experience how useful and effective SCRUM is when working in a team.
 - I also learned how to efficiently use git commands to control my code version and GitHub to collaborate with other team members work.
 - I am more open to talk to everyone in a team at what I am stuck at instead of being awkward at sharing my flaws.
- Shin Hyung Lee:
 - React and Github were new skills I learned from this project. I had a little bit of JavaScript knowledge, but it still felt like a pretty new framework to me. My teammates helped me a lot when I got stuck with certain issues, so I appreciate them.
 - Agile methodology was something I also learned from this project. My teammates were extremely invested in developing our website, and we communicated our daily progress through discord chat. I learned that communicating with the teammates what my needs are is important, because they would know what I am lacking and they can help me.
- Khang Huynh:
 - The first skill I learned is technical writing. During the implementation of the project, I was required to write milestone reports. To explain and provide guidance for users, our team develops technical documentation.
 - Second, I learned how to use Figma for UI design. Third, using ReactJS is the most important skill in this project. It took me a long time to approach this open-source front-end JavaScript library. I used it for the implementation of Learning Hub UI. Fourth, I know using Git and GitHub to collaborate with others.
 - Finally, I learned how to use Selenium on Chrome and Firefox to record one test suite on one complete workflow related to our website. I created test cases for each feature to test our website against various popular browsers and versions.
- Joseph Chang:
 - React JS and GitHub was a new skill I learned and applied to this project. I was somewhat familiar with Javascript but never learned React prior to this project. It was interesting to figure out how the components work and how it can make the code non-repetitive and clean compared to more traditional programming where all the code is on one page without the modularity of components. Github and using the command line was definitely new for me. I've used Github and command line prior to the project but not in the way to collaborate with others.

Need for continuous professional development

(one paragraph per team member)

A.5 By participating in this project, do you feel there is a need for continuous professional development? Provide reasons if you believe so.

- Vy Dinh:

- There is a need for continuous development because we only have a short time to finish this project; thus, there are still many features that need to be improved to complete this project. Also, we don't have much time to test the project on multiple users in different situations, so there might be some problems we need to fix. I believe that our website could be expanded with more useful features.

- Dung Dong:

- Since users would encounter some bugs/errors or bad experiences using our website. There will always be a need for continuous professional development. It is just the first phase of publishing the website, users need time to experience it and give feedback on it. It cannot be perfect just in one development.

- Truong Phu Vu:

- There is definitely a need to further develop our project since it was kind of rushing for us to finish the project. Thus, we only tried to meet the criterias that are required. I would like to spend more time on improving the UI and UX of the website to create such better looking website with smooth animations, better styling of each component appearing on the websites, and better transition between pages. Moreover, I would like to add more functionality for each feature because each feature at the moment is quite basic with no phenomenal functionality to increase the user experience.

- Shin Hyung Lee:

- I could feel that I am lacking in lots of areas during the project. There is a need for continued growth in my programming journey. Working in a team environment was definitely new and gave me motivation to be better because the project result affected all of us. Especially, during the winter break I want to review and learn more of the languages we used for the project and refine my skills.

- Khang Huynh:

- Absolutely, yes, there is a need for continuous professional development. We have one month from discussing and planning to submit a completed project. However, if we want to provide a better user experience while using our website, we need to improve ourselves. When we stop developing ourselves, we will fall behind in this industry. First, we have to upgrade our level when working on Front-end or Back-end Server. Second, project management should be paid more attention to.

Finally, we cannot have a perfect product in just one advancement. We will improve our website and we hope to publish better versions in the future.

- Joseph Chang:
 - I will certainly keep on learning. One of the reasons why I want to become an engineer is that the industry is always evolving. Which means that the learning never stops and there's always a challenge. Maybe not fun all the time but I will never be content or bored at my job. I believe I will continue to learn React JS because I can see the potential use for it. Definitely will learn Figma as well since lots of companies use it. Will probably spend more time with Bootstrap since it's been a while since I really utilized it.

Website implementation and public web hosting preparation process

MongoDB Atlas Cloud Database user account

Database: [SQL4U] [GYHpCzypi18yGqKG]

Database Connection URI:

`mongodb+srv://SQL4U:GYHpCzypi18yGqKG@sql4u.mxnvrm9.mongodb.net/sql4u?retryWrites=true&w=majority`

Connected using MongoDB Compass:

The screenshot shows the MongoDB Compass interface. On the left, there's a sidebar with 'Saved connections' and 'Recents' sections listing various MongoDB connections. The main area is titled 'New Connection' with the sub-section 'Connect to a MongoDB deployment'. It contains a 'URI' input field containing a connection string, an 'Edit Connection String' button, and a 'FAVORITE' button. Below the URI is a 'Save & Connect' button and a 'Connect' button. To the right of the main area, there's a light blue sidebar with instructions for new users, a 'CREATE FREE CLUSTER' button, and links for finding and formatting connection strings.

The screenshot shows the MongoDB Compass interface with the database 'sql4u.mxnvrm9....' selected. The left sidebar shows the database structure with collections like 'courses', 'interviews', 'projects', 'qnas', and 'tutorials'. The main area is titled 'Collections' and lists these collections with their respective storage sizes, document counts, average document sizes, index counts, and total index sizes. A 'Sort by' dropdown is set to 'Collection Name'.

Collection	Storage size:	Documents:	Avg. document size:	Indexes:	Total index size:
courses	24.56 kB	14	685.00 B	1	36.86 kB
interviews	24.56 kB	39	318.00 B	1	36.86 kB
projects	98.30 kB	23	7.73 kB	1	36.86 kB
qnas	20.48 kB	13	303.00 B	1	36.86 kB
tutorials	110.59 kB	129	1.38 kB	1	36.86 kB

Frontend server: AWS Amplify

1. Bought a google domain name.
2. Pushed source code to github.
3. Hosted web app using AWS Amplify

- a. Connect to the github account and choose a github repo for hosting.
- b. Specify branch to host
- c. Configure build script. Since our github repo included both backend and frontend server, we had to specify the path to run the frontend server in the build script.

File: amplify.yml

```

1  version: 1
2  frontend:
3    phases:
4      preBuild:
5        commands:
6          - cd frontend
7          - npm ci
8      build:
9        commands:
10         - npm run build
11      artifacts:
12        baseDirectory: frontend/build
13        files:
14          - '**/*'
15      cache:
16        paths:
17          - node_modules/**/*
18

```

4. Setup Environment Variables for the server.
5. Changed the domain name to our Google Domain:
 - a. Created a CNAME record given by AWS Amplify in Google Domain to verify the owner of the domain.
 - b. Migrate service of our domain to AWS Route 53.
6. The website is automatically re-deploy/rebuilt if there are new commits in the github repository that connected to AWS Amplifier earlier.

Backend Server: Elastic Beanstalk AWS

1. Installed AWS CLI and EB CLI.
2. Followed the instruction of documentation of AWS on deploying Node.js Application to EB: [Deploying Node.js applications to Elastic Beanstalk - AWS Elastic Beanstalk \(amazon.com\)](#)
 - a. **git checkout -b BEdeployment** ⇒ checkout to a branch that is designated for deploying purpose only.
 - b. **cd backend** ⇒ go to the folder backend server.
 - c. **eb init --platform node.js --region us-west-1**
 - d. **eb create --sample node-express-env**
 - e. **git add .**

- f. **git commit -m "Deploying"**
 - g. **eb deploy**
3. Repeat step 2.e → 2.g for re-deploying/ updating backend server. (Have to be on the branch that is designated for deploying, BEdeployment in this case)
 4. Setup Environment Variables for the server.
 5. Enabled encrypted connection through SSL Certificate to handle HTTPS requests.
 - a. Request ACM, Amazon Certificate Manager, for our domain and subdomains.
 - b. Setup SSL Certificate for the server through Load Balancer in EB AWS by setting up a listener to HTTPS requests at port 443 and redirect the request to port 80.
 6. Created an A record in Route 53 to make our subdomain point to the backend server.

Hosted public website

Homepage: <https://www.sql4u.dev/>

Backend Server/ API: <https://api.sql4u.dev>

All endpoints of the frontend server:

```
<Routes>
  <Route element={<PrivateRoutes />}>
    <Route element={<CoursesCard />} path='/' />

    <Route path='/interview' element={<InterviewQuestion />} />
    <Route path='/interview/:id' element={<InterviewQuestionForm />} />

    <Route path='/catalog' element={<CoursesCard />} />
    <Route
      path='/catalog/:courseId/project'
      element={<RecommendProject />}
    />
    <Route
      path='/catalog/:courseId/project/:projectId'
      element={<ProjectPage />}
    />
    <Route
      path='/catalog/:courseId/project/add/:id'
      element={<ProjectForm />}
    />
    <Route
      path='/catalog/:courseId/tutorial/add/:id'
      element={<TutorialForm />}
    />
    <Route path='/catalog/addOrEdit/:id' element={<CourseForm />} />
    <Route path='/catalog/:courseId' element={<IndividualCourse />} />
    <Route
      path='/catalog/:courseId/tutorial/:tutorialId'
      element={<TutorialPage />}
    />

    <Route path='/qna' element={<QnA />} />
    <Route path='/qna/:id' element={<QuestionAndAnswer />} />
    <Route path='/qna/edit/:id' element={<CreatePost />} />

    <Route path='/hub' element={<LearningHub />} />
  </Route>
  {/* <Route path='/' element={<Dashboard />} /> */}
  <Route path='/about' element={<About />} />
  <Route path='/contact' element={<Contact />} />

  <Route path='/logout' element={<Logout />} />
  <Route path='/profile' element={<Dashboard />} /> You, yesterday
  <Route path='/login' element={<Login />} />
  <Route path='/register' element={<Register />} />
```

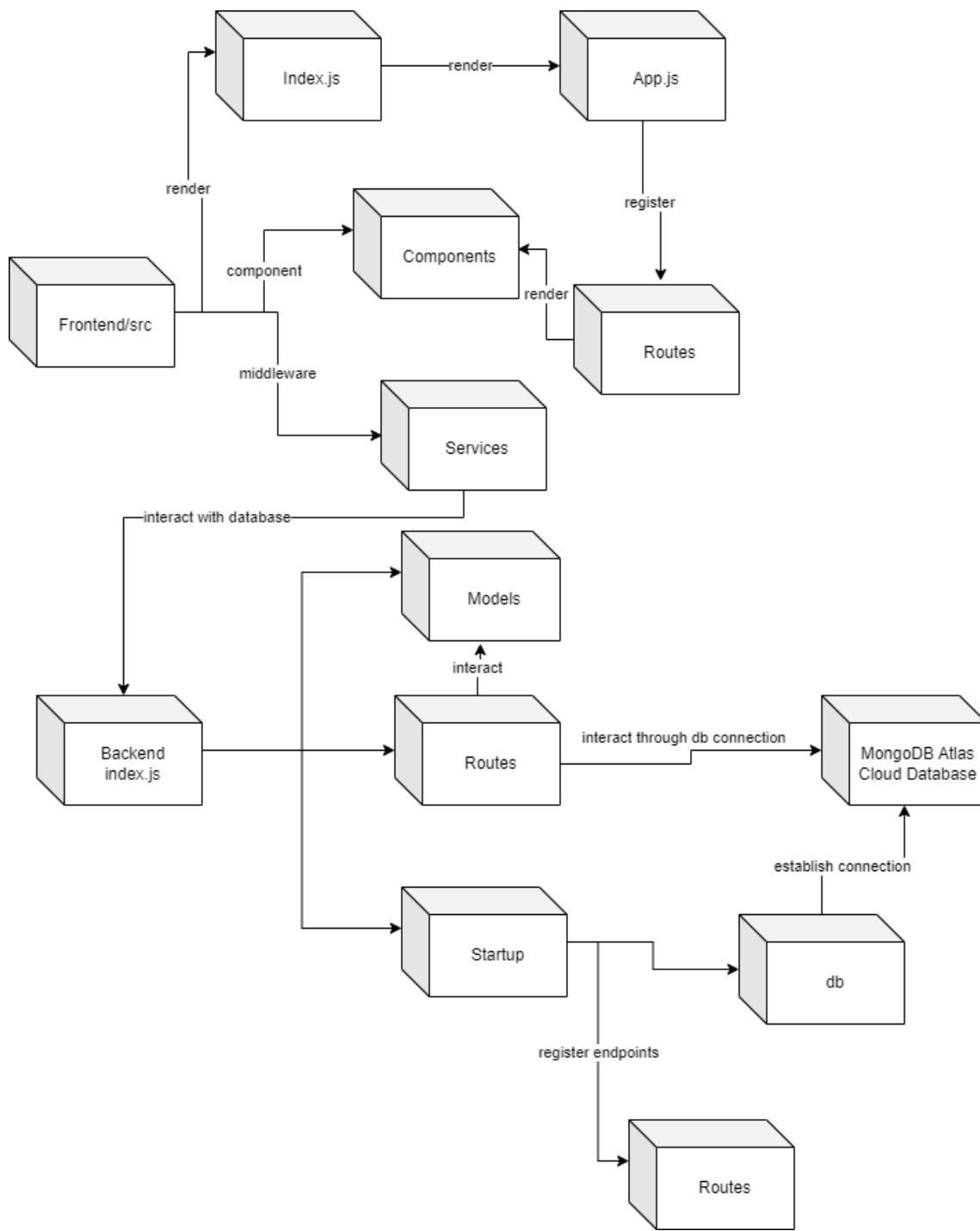
All endpoints of the backend server:

```
app.use(express.json());
app.use("/api/users", user);
app.use("/api/auth", auth);
app.use("/api/project", project);
app.use("/api/course", course);
app.use("/api/tutorial", tutorial);
app.use("/api/qna", qna);
app.use("/api/interview", interview);
```

Show its main components and how they rendered your dynamic website (illustrate with a diagram)

Taxonomy

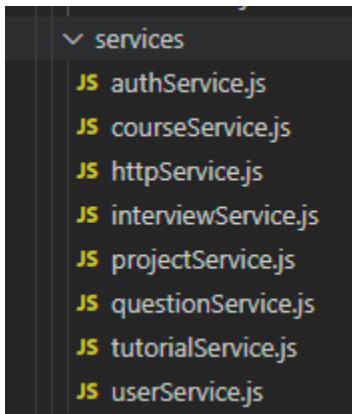
Diagram:



Frontend:

- In charge of rendering contents dynamically based on data exposed via API calls through HTTP services.

- Each feature has its own service acts as a middleware:



Service will be in charge of sending HTTP requests to the backend server to interact with the database.

- App.js register endpoints for features.
- Each component in Components is responsible for a registered endpoint.

Backend:

- In charge of communicating with the MongoDB database hosted on MongoDB Atlas.
- Startup/db: establish connection between
- Startup/Routes: provide endpoints for HTTP requests. Each feature has its own main endpoint:

```
app.use(express.json());
app.use("/api/users", user);
app.use("/api/auth", auth);
app.use("/api/project", project);
app.use("/api/course", course);
app.use("/api/tutorial", tutorial);
app.use("/api/qna", qna);
app.use("/api/interview", interview);
```

- Routes: Handle insert, edit, delete, retrieve documents in a remote database with POST, PUT, GET, DELETE requests.
- Models: Create schema for collection in MongoDB database.
- The front-end server will call these API endpoints to get/insert/edit/delete data.

Bootstrap

- In order to use Bootstrap, we included it in our package.json

```
{  
  "name": "sql4u",  
  "version": "0.1.0",  
  "private": true,  
  "dependencies": {  
    "@testing-library/jest-dom": "^5.16.5",  
    "@testing-library/react": "^13.4.0",  
    "@testing-library/user-event": "^13.5.0",  
    "axios": "^1.1.3",  
    "bootstrap": "^5.2.2",  
    "chart.js": "^3.9.1",  
    "font-awesome": "^4.7.0",  
    "joi": "^17.7.0",  
    "jwt-decode": "^3.1.2",  
    "lodash": "^4.17.21",  
    "mdb-react-ui-kit": "^5.0.0",  
    "primeicons": "^6.0.1",  
    "primereact": "^8.7.2",  
    "query-string": "^7.1.1",  
    "react": "^18.2.0",  
    "react-chartjs-2": "^4.3.1",  
    "react-dom": "^18.2.0",  
    "react-icons": "^4.6.0",  
    "react-router-dom": "^6.4.3",  
    "react-scripts": "5.0.1",  
    "react-toastify": "^9.1.1",  
    "web-vitals": "^2.1.4"  
  },  
}
```

- Index.js will import Bootstrap

21 lines (18 sloc) | 656 Bytes

```
1 import React from "react";
2 import ReactDOM from "react-dom/client";
3 import "./index.css";
4 import App from "./App";
5 import reportWebVitals from "./reportWebVitals";
6 import "bootstrap/dist/css/bootstrap.css";
7 import "font-awesome/css/font-awesome.css";
8
9 import { BrowserRouter } from "react-router-dom";
10
11 const root = ReactDOM.createRoot(document.getElementById("root"));
12 root.render(
13   <BrowserRouter>
14     <App />
15   </BrowserRouter>
16 );
17
18 // If you want to start measuring performance in your app, pass a function
```

- In each of React components where you want to use the button from Bootstrap, you want to import Button from index.js

199 lines (192 sloc) | 6.68 KB

```
1 import React, { Component } from "react";
2 // import NavBar from './navbar';
3 // import { FontAwesomeIcon } from "@fortawesome/react-fontawesome";
4 import "./dashboard.css";
5 import qaImage from "./images/qa3.png";
6 import DashboardSection from "./dashboardSection";
7 import DashboardBox from "./dashboardBox";
8 import DashboardSideBar from "./dashboardSideBar";
9 import Button from ".../index.js";
10 import { useState, useEffect } from "react";
11 import DashboardChart from "./dashboardChart";
12 import * as userService from ".../services/userService";
13 import * as courseService from ".../services/courseService";
14 import shuffle from "lodash/shuffle";
15 // import "bootstrap/dist/css/bootstrap.min.css";
16 // import * as Poop from './dashboardBox';
```

- To implement a Button, you can call the component directly from the HTML code

```

</div>
<div className="dashboardQuestions">
  <DashboardSection title={<a href="/qna">QUESTIONS</a>}>
    <div className="dashboardQuestionsBox">
      <DashboardBox
        title={"Any Questions?"}
        // bgColor="blue"
        icon={<img src={qaImage} height="80" width="100" />}
        textContent={"You can post/find questions here:"}
        button={
          <a href="/qna">
            <button
              type="button"
              className="btn btn-primary btn-lg btn-block"
            >
              Ask!
            </button>
          </a>
        }
    
```

- To lookup how to implement Bootstrap, you can navigate to <https://getbootstrap.com/docs/5.2/getting-started/introduction/>
- Select any of its features, in our case, we used the button component. You can find all the necessary styles and how to use the button in this documentation

Images
Tables
Figures

Forms

Overview
Form control
Select
Checks & radios
Range
Input group
Floating labels
Layout
Validation

Components

Accordion
Alerts
Badge
Breadcrumb
Buttons
Button group
Card
Carousel
Close button
Collapse
Dropdowns
List group

Buttons

[View on GitHub](#)

Use Bootstrap's custom button styles for actions in forms, dialogs, and more with support for multiple sizes, states, and more.



if (you.writeArticle()) {
we.sendSats(you) }

ads via Carbon

Examples

Bootstrap includes several predefined button styles, each serving its own semantic purpose, with a few extras thrown in for more control.

Primary Secondary Success Danger Warning Info Light Dark Link

HTML



```
<button type="button" class="btn btn-primary">Primary</button>
<button type="button" class="btn btn-secondary">Secondary</button>
<button type="button" class="btn btn-success">Success</button>
<button type="button" class="btn btn-danger">Danger</button>
<button type="button" class="btn btn-warning">Warning</button>
<button type="button" class="btn btn-info">Info</button>
<button type="button" class="btn btn-light">Light</button>
<button type="button" class="btn btn-dark">Dark</button>
```

Font-Awesome

- Start by adding font-awesome to package.json

main ➔ SQL4U / frontend / package.json

Truong Phu Vu added footer containing contact and about, fixed some styling

2 contributors

53 lines (53 sloc) | 1.21 KB

```
1  {
2    "name": "sql4u",
3    "version": "0.1.0",
4    "private": true,
5    "dependencies": {
6      "@testing-library/jest-dom": "^5.16.5",
7      "@testing-library/react": "^13.4.0",
8      "@testing-library/user-event": "^13.5.0",
9      "axios": "^1.1.3",
10     "bootstrap": "^5.2.2",
11     "chart.js": "^3.9.1",
12     "font-awesome": "^4.7.0",
13     "joi": "^17.7.0",
14     "jwt-decode": "^3.1.2",
15     "lodash": "^4.17.21",
16     "mdb-react-ui-kit": "^5.0.0",
17     "primeicons": "^6.0.1",
18     "primereact": "^8.7.2",
19     "query-string": "^7.1.1",
20     "react": "^18.2.0",
21     "react-chartjs-2": "^4.3.1",
22     "react-dom": "^18.2.0",
23     "react-icons": "^4.6.0",
24     "react-router-dom": "^6.4.3",
25     "react-scripts": "5.0.1",
26     "react-toastify": "^9.1.1",
27     "web-vitals": "^2.1.4"
28   },
```

- Import to index.js

```

21 lines (18 sloc) | 656 Bytes

1 import React from "react";
2 import ReactDOM from "react-dom/client";
3 import "./index.css";
4 import App from "./App";
5 import reportWebVitals from "./reportWebVitals";
6 import "bootstrap/dist/css/bootstrap.css";
7 import "font-awesome/css/font-awesome.css";
8
9 import { BrowserRouter } from "react-router-dom";
10
11 const root = ReactDOM.createRoot(document.getElementById("root"));
12 root.render(
13   <BrowserRouter>
14     <App />
15   </BrowserRouter>
16 );
17
18 // If you want to start measuring performance in your app, pass a function

```

- From React JS component, you can code directly with font-awesome's className

```

        <a href={`/catalog/${courses[z]._id}`}>{courses[z].name}</a>
      }
    />
</div>
<div className="dashboardMyCoursesBoxBottomRight">
  <DashboardBox
    icon={
      <i className="fa fa-graduation-cap" aria-hidden="true"></i>
    }
    title={


```

- Font-Awesome has a clear instruction on how to use their icons directly on the HTML file



After you get up and running, you can place Font Awesome icons just about anywhere with the `<i>` tag:

`fa-graduation-cap`

```
<i class="fa fa-graduation-cap" aria-hidden="true"></i>
```

Note: to improve web accessibility, we recommend using `aria-hidden="true"` to hide icons used purely for decoration.

i Looking for more? Check out the [examples](#).

A
l
e
d
p
A:
ad

React-icons

- Start by installing “react-icons” by using npm

```
npm install react-icons --save
```

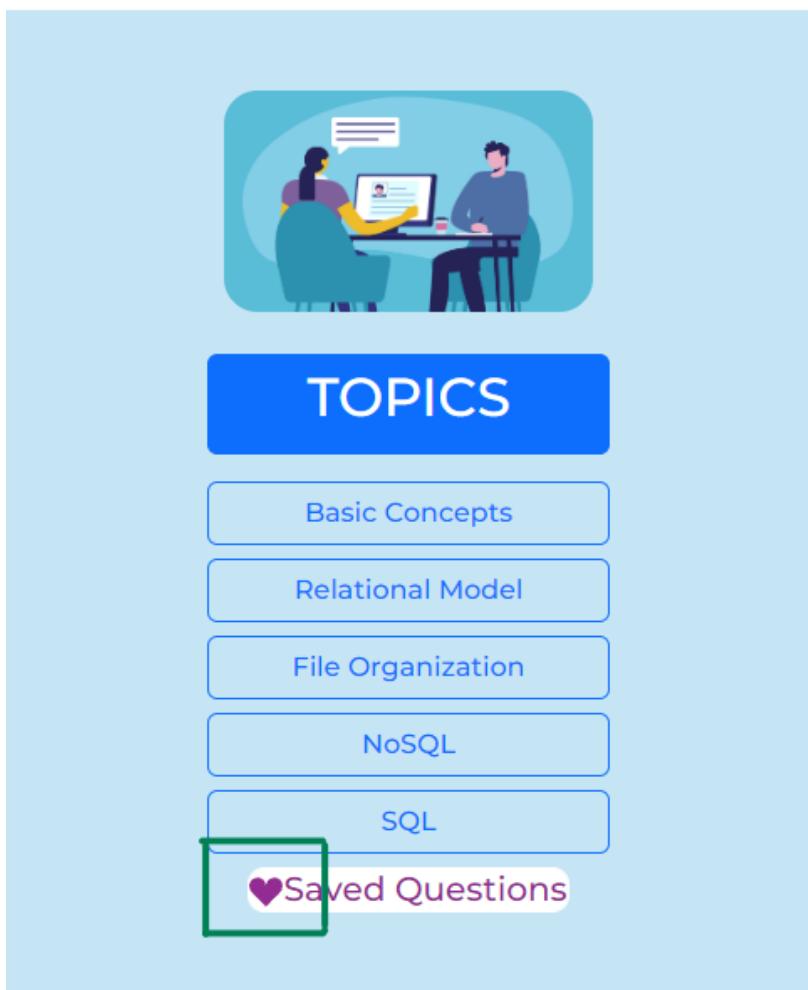
- Reference the react-icons doc to further understand how it is used through:
 - <https://react-icons.github.io/react-icons/>
- To use react-icons within React’s components, firstly, we need to import the desired components from “react-icons/<<component’s folder name>>”. The folder name is usually located as a prefix of the component.

```
3 import { BsSuitHeartFill } from "react-icons/bs";
```

- Within the React, we just need to use the currently installed icon as a Component.

```
137 </div>
138     <button
139         onClick={() => this.handleTopicChange("saved")}
140         className="sidebar-common-questions-btn"
141     >
142         <BsSuitHeartFill />
143         Saved Questions
144     </button>{" "}
145 </div>
```

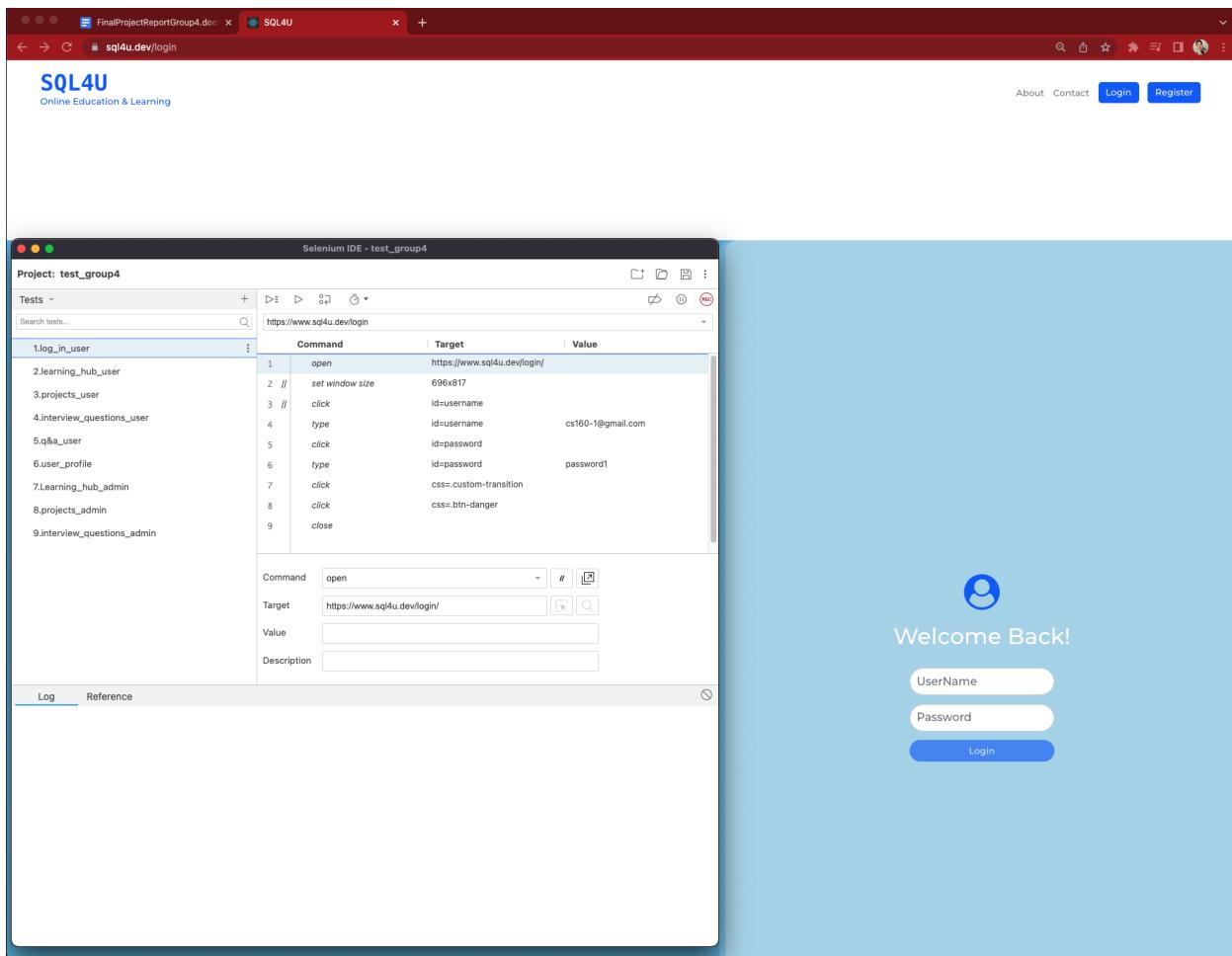
- After finishing the step above, the icon will then appear on the website.



Website Testing

Before the test run begins,

Selenium



During the test run,

The screenshot shows the SQL4U web application interface. At the top, there's a navigation bar with links for 'About', 'Contact', 'Logout', and 'User'. Below the navigation is a menu bar with 'Learning Hub', 'Recommended Projects', 'Interview Question', and 'Q&A'. The main content area features a Selenium IDE window titled 'Selenium IDE - test_group4'. The IDE displays a list of test steps for a project named 'test_group4'. The steps include actions like 'set window size', 'click', 'type', and 'mouse over' on various UI elements. The log panel at the bottom shows the execution details of these steps, with each step followed by its status ('OK') and timestamp. The URL in the browser's address bar is <https://www.sql4u.dev/projectLandingPage/6385bc14870e807b4cfcd16/project>.

Command	Target	Value
2 ✓ set window size	696x17	
3 ✓ click	id=username	
4 ✓ type	id=username	cs160-1@gmail.com
5 ✓ click	id=password	
6 ✓ type	id=password	password1
7 ✓ click	css:custom-transition	
8 ✓ click	css=nav-item:nth-child(2) .badge	
9 ✓ click	css=items:nth-child(4) > .btn	
10 mouse over	linkText:Shin Hyung Lee: shin hyung.lee@sjtu.edu	

Log:

```

13. click on css=btn-danger OK
14. close OK
'2.learning_hub_user' completed successfully
Running '3.projects_user'
1. open on https://www.sql4u.dev/login/ OK
2. setWindowSize on 696x17 OK
3. click on id=username OK
4. type on id=username with value cs160-1@gmail.com OK
5. click on id=password OK
6. type on id=password with value password1 OK
7. click on css:custom-transition OK
8. click on css=nav-item:nth-child(2) .badge OK
9. click on css=items:nth-child(4) > .btn OK
10. mouseOver on linkText:Shin Hyung Lee: shinhyung.lee@sjtu.edu
  
```

The screenshot shows a dual-pane interface. On the left, a sidebar titled "----TOPIC----" lists categories like "Basic Concepts", "Relational Model", etc. Below it are user profiles: "cs160-1@gmail.com", "dung.dong@sjsu.edu", and "dung.dong@sjsu.edu". On the right, a main area displays a Selenium IDE session titled "Project: test_group4". The session is executing a script to log in to the SQL4U login page. The script includes commands for logging in, navigating to the learning hub, and interacting with a badge. The log pane shows the execution details, including the sequence of steps and their timestamps.

Step	Action	Target	Value	Timestamp
1	open	https://www.sql4u.dev/login		23:39:52
2	setWindowSize	696x817		23:39:52
3	click	id=username	password1	23:39:52
4	type	id=password	password1	23:39:58
5	click	css=custom-transition		23:40:01
6	click	css=nav-item:nth-child(4) > .badge		23:40:04
7	click	id:title	Basic Concept 1	23:40:07
8	click	id:topic		23:40:11
9	select	label=Basic Concepts		23:40:14
10	click	id:description		23:40:17
11	type	id:description	hello ngay moi	23:40:20
12	click	id:topic		23:40:23
13	click	label=Basic Concepts		23:40:27
14	type	id:description	hello ngay moi	23:40:30

Selenium IDE - test_group4*

Project: test_group4*

Executing ~

Topic: No

Category: Test

Topic: To

Sub-Topic: 8.projects_admin

Block: 9.interview_questions_admin

Runs: 6 Failures: 0

Log Reference

```

"6.user_profile" completed successfully
Running "7.Learning_hub_admin"
1. open https://www.sql4u.dev/login/ OK
2. setWindowSize on 1386x821 OK
3. click on id=username OK
4. type on id=username with value sql4u@gmail.com OK
5. click on id=password OK
6. type on id=password with value SQL4U@ OK
7. sendKeys on id=password with value ${KEY_ENTER} OK
8. click on linkText= Add New Course OK
9. click on id=topic OK
10. select on id=topic with value label=NoSQL OK
11. click on id=name OK
12. type on id=name with value Test98 OK

```

Test execution speed: https://www.sql4u.dev/0

Command	Target	Value
3. ✓ click	id=username	
4. ✓ type	id=username	sql4u@gmail.com
5. ✓ click	id=password	
6. ✓ type	id=password	SQL4U@
7. ✓ send keys	id=password	\${KEY_ENTER}
8. ✓ click	linkText= Add New Course	
9. ✓ click	id=topic	
10. ✓ select	id=topic	label=NoSQL
11. ✓ click	id=name	
12. type	id=name	Test98

Command Target Value Description

At the end of Test completion,

The screenshot shows a browser window with the SQL4U website. The main content area displays a Selenium IDE interface for a project named 'test_group4'. The test plan includes steps for logging in, navigating through various user and admin sections, and interacting with database management and NoSQL topics. To the right of the IDE, two recommended projects are listed:

- Database III: Database Management** by Adam Wilbert (5.0 stars, 2h 14m)
- SQL vs. NoSQL: Which Type Is Right for You?** by David Jones-Gilardi (5.0 stars, 0h 29m)

Testing Platform Information

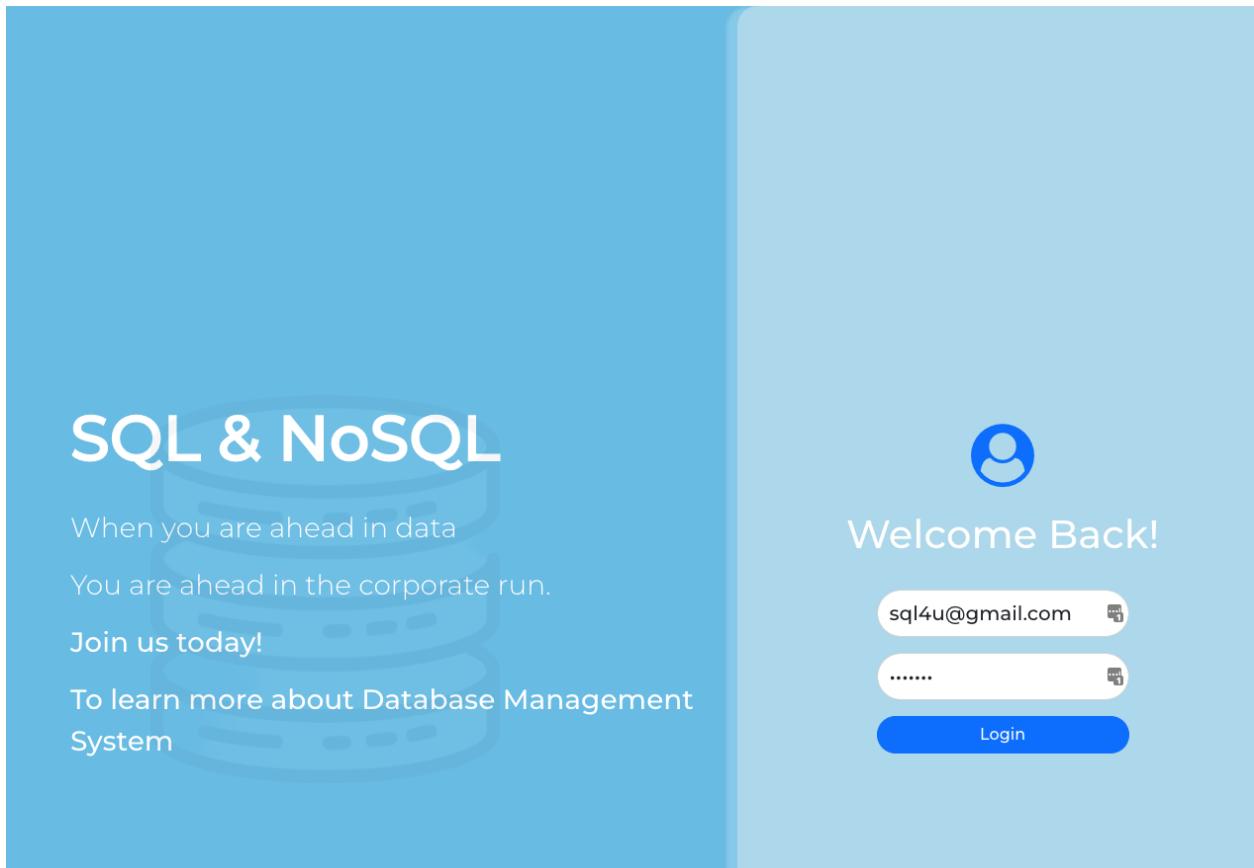
Client OS	Client Browser Versions	Server OS	Outcome
macOS Big Sur Version 11.5.2	Google Chrome Version 108.0.5359.94 (Official Build) (x86_64)	64-bit Amazon Linux 2/5.6.1	No Issues
macOS Big Sur Version 11.5.2	Firefox 107.0.1 (64 bit)		No Issues

macOS Big Sur Version 11.5.2	Microsoft Edge Version 108.0.1462.46 (Official build) (x86_64)		No Issues
Windows 10	Firefox 107.0.1 (64 bit)		No Issues
Windows 10	Google Chrome Version 107.0.5304.107 (Official Build) (x86_64)		No Issues

For each of your implemented proposed features
A typical flow of your website with narrated snapshots

Landing page: (Login)

- Landing page where the user will see first going to the website



Register page:

- The page where the user needs to register and create an account in order to login

SQL & NoSQL

When you are ahead in data
You are ahead in the corporate run.

Join us today!

To learn more about Database Management System

After login: (Learning hub)

- First page the user will see after logging in.

TOPIC

All

Basic Concepts

Relational Model

File Organization

NoSQL

SQL



Database I: Intro to Databases

★★★★★ (5.0)

Adam Wilbert
3h 21m

[GO!](#)



Database II: Data Structures

★★★★★ (5.0)

Adam Wilbert
2h 55m

[GO!](#)



Database III: Database Management

★★★★★ (5.0)

Adam Wilbert
2h 14m

[GO!](#)



Learning Relational



Relational Databases



File Organization in DBMS |

Learning Hub: Choose a Topic

- Users can choose which topic they would like to view by choosing one of the topics button on top

TOPIC

All

Basic Concepts

Relational Model

File Organization

NoSQL

SQL



Database I: Intro to Databases

★★★★★ (5.0)

Adam Wilbert
3h 21m

[GO!](#)



Database II: Data Structures

★★★★★ (5.0)

Adam Wilbert
2h 55m

[GO!](#)



Database III: Database Management

★★★★★ (5.0)

Adam Wilbert
2h 14m

[GO!](#)

Course Catalog:

- Once the user clicks into a topic, they will be directed to a course catalog page where it lists all the relevant courses.
- User can choose between going into a specific learning content or go to recommended projects

[Go back](#)

Database I: Intro to Databases

1. Introduction
2. Database Core Concepts
3. How to Set Up a Database Playground?
4. Graphical Interfaces
5. Create Table Objects
6. Add Data to a Table
7. Retrieve Information
8. Manipulate Data
9. References
10. Projects

We Picked Some Projects For You To Practice Your Skills

[Let's Go >>](#)

Learning Content:

- When the user clicks into the learning content links, they'll be directed to the learnings page

[Go Back](#)

Introduction

An introduction to database

What Is a Database?

--> A database is an organized collection of material that has been arranged and is often kept electronically in a computer system. A database management system often oversees a database (DBMS). The term "database system," which is frequently abbreviated to "database," refers to the combination of the data, the DBMS, and the applications that are connected to it.

--> To facilitate processing and data querying, the most popular types of databases currently in use typically model their data as rows and columns in a set of tables. The data may then be handled, updated, regulated, and structured with ease. For writing and querying data, most databases employ structured query language (SQL).

What you should know?

--> No prior database experience required

Recommended Project:

- The recommended project feature can be accessed through the navigation bar by clicking to the “Recommended Projects” button.

TOPIC

All

Basic Concepts

Relational Model

File Organization

NoSQL

SQL



Database I: Intro to Databases

★★★★★ (5.0)



Adam Wilbert

3h 21m

[See Projects](#)



Database II: Data Structures

★★★★★ (5.0)



Adam Wilbert

2h 55m

[See Projects](#)



Database III: Database Management

★★★★★ (5.0)



Adam Wilbert

2h 14m

[See Projects](#)



Learning Relational



Relational Databases



File Organization in DBMS |

- When the user clicks into “See Projects” button, they’ll be directed to the recommended projects page.
- The recommended projects page will show a list of projects that are relevant to the user. Once the user completes the project, there will be a check mark displayed.

Database I: Intro to Databases

Project #1 - SQL - **BEGINNER**

Project #2 BUFFER POOL - **INTERMEDIATE**

Project #3 - HASH INDEX - **PROFESSIONAL**

Project #4 - QUERY EXECUTION - **PROFESSIONAL**

[Go Back](#)



- The content of each project will be displayed on screen once a user click to individual project title.

Project #1 - SQL

OVERVIEW

The first homework is to construct a set of SQL queries for analyzing a dataset that will be provided to you. For this, you will look into IMDB data. This homework is an opportunity to: (1) learn basic and certain advanced SQL features, and (2) get familiar with using a full-featured DBMS, SQLite, that can be useful for you in the future.

This single-person project will be completed individually (i.e., no groups).

SPECIFICATION

The homework contains 10 questions in total and is graded out of 100 points. For each question, you will need to construct a SQL query that fetches the desired data from the SQLite DBMS. It will likely take you approximately 6-8 hours to complete the questions.

PLACEHOLDER FOLDER

Create the placeholder submission folder with the empty SQL files that you will use for each question:

```
$ mkdir placeholder
$ cd placeholder
$ touch q1_sample.sql \
q2_uncommon_type.sql \
q3_tv_vs_movie.sql \
q4_old_is_not_gold.sql \
q5_percentage.sql \
q6_dubbed_smash.sql \
q7_imdb_250.sql \
q8_number_of_actors.sql \
q9_movie_genres.sql
```

- There is a second way to access the recommended projects which is through learning hub's courses. When a user finishes studying a course, at the end of the page, there will be a button which navigate the user to the projects for that certain course.

[Go back](#)

Database I: Intro to Databases

1. Introduction
2. Database Core Concepts
3. How to Set Up a Database Playground?
4. Graphical Interfaces
5. Create Table Objects
6. Add Data to a Table
7. Retrieve Information
8. Manipulate Data
9. References
10. Projects

We Picked Some Projects For You To Practice Your Skills

[Let's Go >>](#)

Interview Questions:

- Interview questions page will list questions that are helpful in an interview related to the database.
- Users can choose a specific topic from the left side's navigation bar. There are 5 different topics to choose from. (Basic Concepts, Relational Model, File Organization, SQL, NoSQL)

The screenshot shows the 'Interview Questions' section of the SQL4U website. On the left, there's a sidebar titled 'TOPICS' with icons for two people at a desk and a computer monitor. Below this are five categories: 'Basic Concepts', 'Relational Model', 'File Organization', 'NoSQL', and 'SQL'. At the bottom of the sidebar is a purple button labeled 'Saved Questions' with a heart icon. The main content area has three questions listed in boxes:

- What are NoSQL databases?** (Answer +) [Save Question](#)
- What are the different types of NoSQL databases?** (Answer +) [Save Question](#)
- Why would we need to use NoSQL?** (Answer +) [Save Question](#)

Interview Questions: (Answer)

- Users can click the answer button to reveal the question to the interview questions



Interview Questions

What are NoSQL databases?

Answer -

A NoSQL database provides a mechanism for storage and retrieval of data that is modeled in means other than the tabular relations used in relational databases (like SQL, Oracle, etc.).

Save Question

What are the different types of NoSQL databases?

Answer +

Save Question

Why would we need to use NoSQL?

TOPICS

- Basic Concepts
- Relational Model
- File Organization
- NoSQL
- SQL

Saved Questions

Interview Questions: (Saved Questions)

- Users can save the questions then click on saved questions button on the left to view them



Interview Questions

What are the differences between a DBMS and RDBMS?

Answer +

Unsave Question

What are NoSQL databases?

Answer +

Unsave Question

TOPICS

- Basic Concepts
- Relational Model
- File Organization
- NoSQL
- SQL

Saved Questions

Interview Questions: (Unsave Question)

- Users can unsave the questions from “Saved Questions” list by clicking on unsave question button on the left bottom corner

The screenshot shows a mobile application interface. On the left, there is a sidebar titled "TOPICS" with a list of categories: Basic Concepts, Relational Model, File Organization, NoSQL, SQL, and a pink-highlighted "Saved Questions". Above the sidebar is a small illustration of two people at a desk. The main content area has a blue header "Interview Questions". It contains two questions in cards: "What are the differences between a DBMS and RDBMS?" and "What are NoSQL databases?". Each card has a "Unsave Question" button at the bottom left and an "Answer + " button at the top right.

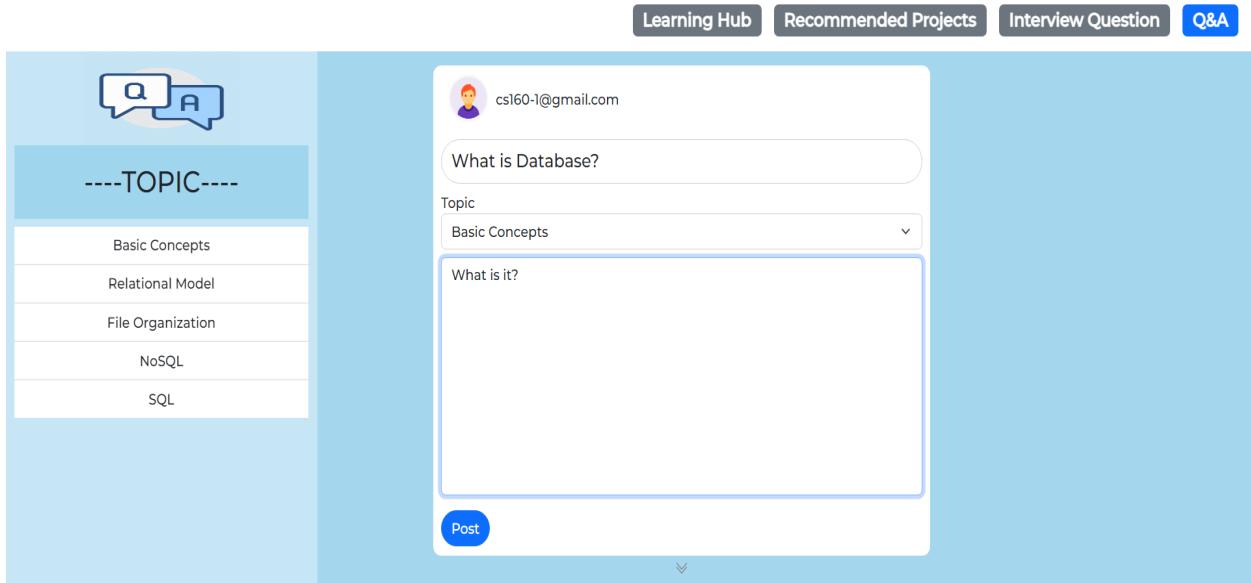
Q&A:

- Q&A page will have a form to ask questions and will also have list of previously asked questions

The screenshot shows a user interface for asking a question. At the top, there are four tabs: 'Learning Hub', 'Recommended Projects', 'Interview Question', and 'Q&A'. The 'Q&A' tab is highlighted with a blue background. Below the tabs, there is a sidebar on the left with a 'Topic' section containing a list of topics: 'Basic Concepts', 'Relational Model', 'File Organization', 'NoSQL', and 'SQL'. To the right of the sidebar is a main form area. The form includes fields for 'Question' (with placeholder text 'cs160-1@gmail.com'), 'Topic' (with a dropdown menu), and 'Description' (a large text area). At the bottom of the form is a 'Post' button.

Q&A: (Post a question)

- Users can post a new question



Q&A: (Reply to a question)

- Users can view the questions on the page and reply with answers



cs160-1@gmail.com



Topic: Basic Concepts

Title: What is a database?

Description:

What is it?



This is an answer

Reply

Delete



cs160-1@gmail.com

Q&A: (Delete a comment)

- Users can delete their own replies

The screenshot shows a user profile interface. At the top, there's a header bar with tabs: Learning Hub, Recommended Projects, Interview Question, and Q&A. Below the header, a blue header bar displays a user icon and the email address `cs160-1@gmail.com`. To the right of the email are two small icons: a gear and a trash can. The main content area has a white background. It contains a post with the following details:

- Topic: Basic Concepts
- Title: What is a database?
- Description: What is it?

Below the post is a large, empty text input field labeled "Reply". Underneath this field are two buttons: a blue "Reply" button and a red "Delete" button. A small downward arrow icon is positioned between the reply field and the buttons. At the bottom of the main content area, there's another blue header bar with a user icon and the email address `cs160-1@gmail.com`, followed by the timestamp `2022-12-09T04:31:47.547Z`. A blue rectangular box contains the text "This is an answer".

User Profile:

- User Profile page where it's a dashboard and they can get an overview of some of the features on the page starting from saved interview questions, finished projects, and courses they've clicked into.

Dashboard

Courses you might interested



Relational Databases
Essential Training



File Organization in DBMS |
Set 3



SQL vs. NoSQL: Which Type
Is Right for You?



Database II: Data Structures

QUESTIONS



Any Questions?

You can post/find questions here:

[Ask!](#)

INTERVIEW QUESTIONS



Questions you've saved

What are the differences
between a DBMS and
RDBMS?

FINISHED PROJECTS



Completed Projects

[Project #1 - SQL](#)

[Project #2 BUFFER POOL](#)

[Project #3 - HASH INDEX](#)



Provide relevant final spec, use cases, and use case diagrams

Feature #1 Interview Questions (Core Requirement C1 and C3)

Interview Questions

- i. Users are allowed to choose a topic
- ii. Answers are hidden at first so users can take time and think about it.
- iii. There is a button to reveal the answer for a question (can be collapsed and expanded)
- iv. Questions can be various (different theoretical topics + technical) - took out coding questions because we were not sure how that would be implemented.
- v. We could possibly apply the progress checking feature here so that users also have a brief report of their understanding.
- vi. Users can save their favorite questions into a list so that they can look for it easier.
- vii. Users can unsave a saved question.

How it will be useful to users

- i. It is a tool for them to check their knowledge on a particular topic after learning and prepare for their future interview.
- ii. Users can test their knowledge on a particular topic and expect what they will be asked in an interview.
- iii. Easily review their favorite questions.
- iii. Knowledge from different database topics are tested to further users' understanding

Why this is Innovative

The purpose of this feature is to let users have better preparation for their interviews in their future on database concepts. This is the place where they can test their knowledge on different topics in DBMS.

A **answer** button is created to make users actually think about the answer so that they can remember it longer and better. A save button is implemented so that users are able to go back to this question easier, without having to look for it among a bunch of questions. **Unsave button was added so users can remove questions from “Saved Questions” list if they wish to.** By default, questions are filtered by their topic, but the user can create a list of their own questions.

URL References

<https://leetcode.com/problems/combine-two-tables/>

Use Case ID	C1-1
Use Case Name	Interview Questions
Created By	Shinhyung Lee
Actor	User, database
Description and Event Flow	<ul style="list-style-type: none"> - Users are asked to choose a topic that they want for their sample questions - A list of questions in the chosen topic will show hidden answers. - Users click the “answer” button to see the answer for a particular problem/question. - They can save it to their favorite questions list. - Users are able to unsave questions if they decide to.
Pre-Conditions	<ul style="list-style-type: none"> - Log in to their account so that they can keep track of their learning - Must be connected to the internet
Post-Conditions	<ul style="list-style-type: none"> - Favorite questions are added to the user's account. - The first-time user will have an empty favorite questions list if he/she does not add questions to it. - If a user unsaved a question, it will be removed from their favorite questions list.
Input	<ul style="list-style-type: none"> - Topic - Saved Questions
Output	<ul style="list-style-type: none"> - Questions within the input topic shown up - Successfully save the question to the database.
Includes	
Notes	-Sign up/Log in is required if the user wants to add questions to favorite questions list

Feature #2 Learning Hub (Core Requirement C1, C2, C3 and C5)

Learning Hub

- i. This feature is a set of pages where users can learn about the technical subject (Database)
- ii. There are two types of users: first-time and returning users
 - _ All users will be placed on the course catalog, where they can access all tutorial courses and an overview of learning progress. They can select their favorite topics and see the relative courses.
 - _ Also, Learning Hub will be supported by Recommended Project.
 - _ Based on what they learned, Learning Hub will provide them with recommended projects. Users will be well-prepared to succeed in landing a new job in this technical subject.
- iii. After users log out, all activities will be kept track by supporting of check progress feature

How it will be useful to users

- i. This feature is the first step for beginners to approach this technical subject (Database)
- ii. It provides enough basic knowledge about databases. Also, it sets the stage for a solid background which helps future development.
- iii. This feature is not only a new place where beginners learn the material but also help professionals to consolidate their knowledge
- iv. Users should be concerned with database design because it is crucial to the consistency, integrity, and accuracy of the data in a database. This feature can improve designing database skills by [providing some courses about it](#).

Why this is Innovative

The main purpose of this feature is to advise users a correct learning roadmap. Basically, traditional pages have some limits to learning content, although there are many different types of databases. The best database for a specific organization depends on how the organization intends to use the data. But most commonly, databases come in two categories, relational and non-relational. By comparing the advantages and disadvantages of each database, users can know which is suitable for their project because a well-designed database contains accurate and up-to-date information for analysis and reporting. In this feature, users will be able to learn two common types of databases. Learning Hub not only provides basic concepts but also is

a database advisor. This feature shows step-by-step how users can approach database management from beginner to expert. Learners can practice some exercises before moving to projects. Learning Hub is received support from recommended projects, which means users can organize their knowledge while gaining an understanding of the database. Finally, users can gain some valuable experience in learning through this feature because the check progress feature will save all history activities and display all materials learners need to finish to complete the course.

URL References

<https://www.oracle.com/database/what-is-database/#link5>

<https://www.w3schools.com/sql/>

<https://www.guru99.com/nosql-tutorial.html>

Use Case ID	C1-2
Use Case Name	Learning Hub
Created By	Khang Huynh
Actor	User, Administrator
Description and Event Flow	<p>A new or returning users want to access Learning Hub</p> <ul style="list-style-type: none"> ▪ User logged in and on the homepage ▪ The first page of this feature is the courses catalog where they can access all tutorial courses and an overview of learning progress ▪ The Course Catalog contains all courses covering the technical subject. Users can filter courses based on their favorite topic. ▪ The first-time user (level: beginner) can start by learning the basic concepts topic ▪ The first-time user (level: professional) can skip the basic concepts and select their favorite topics/courses. ▪ Each course shows a list of tutorials. When users complete all tutorials of this course, they can see a button that leads them to Recommended Projects. Based on what they learned, it will show some projects and help users apply practical knowledge of the database ▪ Also, the progress feature will allow them to check all history activities. Users can see previously completed tabs or the rest to complete the course. <p>The administrator want to add/edit/delete courses</p> <ul style="list-style-type: none"> ▪ The administrator wants to add a new course. <ul style="list-style-type: none"> ○ Click “Add a new course button” next to topics ○ Select the topic for the course ○ Enter a course name, teacher name and total time ○ Click “Submit” to add the course ▪ The administrator wants to edit a course. <ul style="list-style-type: none"> ○ Click the “Edit” button in the course card ○ Change topic, course name, teacher name, or total time ○ Click “Submit” to save the course ▪ The administrator want to delete a course. <ul style="list-style-type: none"> ○ Click the “Delete” button in the course card

Pre-Conditions	<ul style="list-style-type: none"> A new user is created, or a user exists A user is logged in and on the home page An administrator is logged in
Post-Condition s	<ul style="list-style-type: none"> A user can go to the homepage and log out Check progress keeps track of all users' activities An administrator can add/edit/delete courses
Input	<ul style="list-style-type: none"> Users click which courses they want to learn Administrator clicks add new courses Administrator clicks courses they want to edit/delete
Output	<ul style="list-style-type: none"> Users see learning content. Users see recommended projects at the end of each finished course Administrator can see new courses and edit/delete current courses.
Includes	
Notes	(Option) Users without an account can approach this feature, but they do not have permission to access other features

Feature #3 Recommended Projects (Core Requirement C3 & C5)

Recommended Projects

1. A user can refer to the recommended projects listed at the bottom of each subcategory to practice their newly learned skills.
2. A user can utilize a wide range of knowledge they have gained to work on the comprehensive recommended projects.
3. There will be three levels for comprehensive projects such as beginner, intermediate, and advanced.

How it will be useful to users

1. Learning the concept alone might be tiring and uninteresting. Moreover, humans can learn faster if they activate their mind muscles, therefore, working on a project related to the concept they have just learned will create such a strong bond. Thus, the users can clearly understand the concept and retain the information longer.

Why this is Innovative

The purpose of the recommended project feature is to help users orient and visualize their thoughts and knowledge. There will be small projects attached to the bottom of each subcategory. Each subcategory project will ask the user to utilize the fundamental knowledge within that subcategory to finish the work.

Beginners usually can't picture how the information they have just learned will be used properly, especially in coding, therefore, this feature can satisfy the needs of beginners. On the other hand, there are also small projects in each subcategory that require users with professional skills in database to solve. Thus, this feature can satisfy the needs of both beginners and professionals. Not only does this feature provide recommended projects in each subcategory, but it also provides comprehensive projects. Comprehensive projects are the kinds of projects that users must integrate a wide range of knowledge to solve. Thus, users can visualize how the information they have learned is combined as a whole.

URL References

<https://www.freecodecamp.org/learn/2022/responsive-web-design/#learn-html-by-building-a-cat-photo-app>

<https://www.freecodecamp.org/learn/back-end-development-and-apis/#mongodb-and-mongoose>

Use Case ID	C1-3
Use Case Name	Recommended Projects
Created By	Truong Phu Vu
Actor	User
Description and Event Flow	<p>The recommended projects are listed at the bottom of each subcategory and category for user to practice</p> <ol style="list-style-type: none">1. User logs in their account.2. There are two ways to access the Recommended Projects feature:

- a. First way: Through “Learning Hub” feature
- i. User is presented with a “Projects” button at the bottom of each course.
 - ii. The list of projects will appear.
 - iii. User can expand or collapse the Projects div by a button. The project list will be fully extended, there is no button to expand or collapse the div.
 - iv. The projects are displayed in 2 sections: beginners and advanced for subcategory and 3 sections: beginners, intermediate, and advanced for category. Next to each project’s title there will be a level specifically for that project such as Beginner, Intermediate, or, Professional
 - v. A full project description containing prompts will be pulled from the database to render in a new tab of web browser when the user selects a project.
 - vi. In the description, users will be guided step by step to complete the project. After completing the project steps, the user will press the “finished” button to mark that they have finished the project.
 - vii. User submits the screenshot of the project’s results through a submit button if they have finished the project. Users will not submit the screenshot anymore, the motto of the website is self-learning so there is no grading needed.
 - viii. The system will update the isFinished state of that project in database to True. when they click “Finished” button, the system will update the isFinished state of that project in database to true to mark that the user already finished that project. The text of the “finish” button will then chane to “Finished” and the check mark of the project on the previous page will be filled. If the button is not clicked the isFinished state will stay false and the other front end components relate to IsFinished state will also stay the same.
 - ix. User can go back to the previous page by navigating back to the tab whenever they would like to, project submission is not a requirement to leave the page
- b. Second way: Through features vavigation bar

	<ul style="list-style-type: none"> i. When users click to the button “Recommended Projects” button on the feature navigation bar, they will be navigated to the Recommended Projects landing page which shows the list of courses. ii. Users click to the button “See Projects” under each course will lead them to the projects recommended for that course. iii. The event flow stay the same as a.ii to a.ix
Pre-Conditions	<ul style="list-style-type: none"> ▪ A new user is created or a user exists ▪ A user logged in ▪ Every project isFinished state is set to False
Post-Condition s	<ul style="list-style-type: none"> ▪ A screenshot of the projects’ results is submitted through a submit button. There is no need for the screenshot anymore. The system will verify the results of the project through whether the user has clicked the “Finised” button or not ▪ The isFinished state of the project in the database is changed to True. If the current project’s “Finished” button is not clicked, the isFinished state will remain False ▪ The check mark icon will be filled with blue color (it was grey before Finished button is clicked)
Input	<ul style="list-style-type: none"> ▪ A user selects a project
Output	<ul style="list-style-type: none"> ▪ The submit button is clicked. ▪ The isFinshed state of the project is changed to True.
Includes	
Notes	There will be a case the user does not submit their screenshot of the result. The isFinished state will not be changed to True.

Feature #4 Q&A Page (C3 and C5)

Q&A Page

- i. Log-in users are able to create new posts for a question
- ii. Each post will include a title, user’s name, and published date

- iii. Users will be able to delete or edit their posts or reply to other questions
- iv. There will be some default common questions, and answers from each topic

How it will be useful to users

The Q&A Page is a place where users can find solutions for their problems by posting questions. In this feature, users will also be able to discuss the learning topic or problems with other users. Thus, it helps users to understand deeply about the new content and to review their knowledge.

Why this is Innovative

The purpose of Q&A page is to create a good way of self-learning for users. Questioning and discussing what you have just studied is a necessary method for making your learning more effective. After users learn new categories from the learning path section, users can find answers or post questions directly on this website. This website is made only for users who want to explore databases. Thus, users can easily find the answers related to the topic, or they can find classmates to discuss the same problems. Especially, users will be able to see many ideas and contributions from both beginners and professionals. Moreover, users can have some questions beyond their knowledge such as technical problems or career paths. Beginner users are an example of starters who don't know where they should start or what is better for their careers. Thus, they can seek help from professionals with experience in this field. As a result, this page can help users exchange their knowledge and experience. Both professionals and beginners can review and learn more from each other.

URL References

<https://www.tutorialspoint.com/questions/index.php>

Use Case ID	C1-4
Use Case Name	Q&A Page
Created By	Vy Dinh
Actor	Users
Description and Event Flow	<ul style="list-style-type: none"> - Users need to log in to do activities in this page - There will be creating selection for users to start activity - There are some questions displayed on the page with comment box for users to choose to reply - Log-in users can post questions, reply to other questions - Users can only delete their own post or replies

	- A post will contain title, user's name, published date
Pre-Conditions	<ul style="list-style-type: none"> - An account is created - User is signed in - User need zero post to be able to comment on other question - User need one or more posts to be able to delete
Post-Conditions	<ul style="list-style-type: none"> - User will be received email if someone leaves comment on their post
Input	<ul style="list-style-type: none"> - Create selection for users post questions - Reply selection for users post answers - Delete and edit selection
Output	<ul style="list-style-type: none"> - Answers - Notification email
Includes	
Notes	<ul style="list-style-type: none"> -Users need to be signed in to post - Users can reply to any questions - Users can only delete and edit their posts

Feature #5 User Profile (C5)

(Revised to User Profile Dashboard from Dashboard)

User Profile Dashboard

- i. Dashboard feature gives users an overview of their overall learning progress.
- ii. Users can view their course progress percentages, previously viewed interview questions, and completed projects.
- iv. Personalized experience, users can view to individual courses they are currently taking, interview questions that were saved, and finished projects
- vi. Every user will have a different dashboard corresponding to their learning progress, so this will add individualized learning components to the website experience.

How it will be useful to users

Dashboard gives users a comprehensive summary of their learning progress. Learners can track how much they have learned for each course / topic they are enrolled in. They can check completed projects here. This feature is highly individualized for each user.

Why This Is Innovative

By having an individualized Dashboard for each user, the learning process can be less boring for the users. Learning can be arduous, and it is our responsibility as content providers to make the learning journey enjoyable for learners to the best of our ability.

URL References

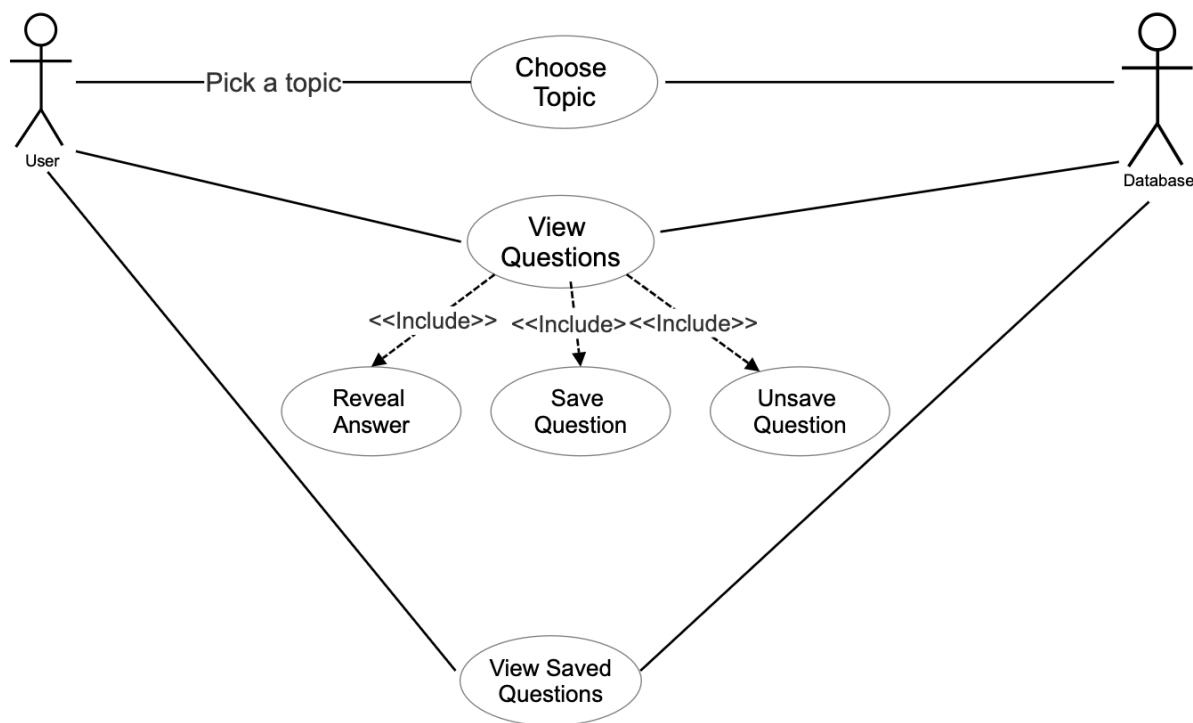
-User needs to be logged in to view this page, so we are attaching an image below

Use Case ID	C1-5
Use Case Name	User Profile Dashboard (This feature is updated from Dashboard)
Created By	Joseph Chang
Actor	Users
Description and Event Flow	<ul style="list-style-type: none"> - When user is logged in, and click on user button on top will see is this “Dashboard” page - Dashboard shows a comprehensive summary of the user’s learning progress - Depending on where the user decides to click on, it will redirect them to the specific page (i.e. Learning Hub, Q & A, Interview Questions)
Pre-Conditions	<ul style="list-style-type: none"> - An account is created - User is signed in - For interview questions to generate on the dashboard, a user must view questions on interview questions page - For finished projects to generate a view on the dashboard, a user must have completed a project - For any projects to be listed in “Completed Projects”, users will have to finish a specific recommended project.
Post-Conditions	<ul style="list-style-type: none"> - User is redirected to individual feature page they clicked on
Input	<ul style="list-style-type: none"> - Click on the button of choice (“Ask”, “Start now”, Interview Questions)
Output	<ul style="list-style-type: none"> - Users will be redirected to the “Q&A” page if they clicked on the “Ask” button. - Users will be redirected to specific courses in Learning Hub.
Includes	
Notes	<ul style="list-style-type: none"> - User needs to be signed in with their information

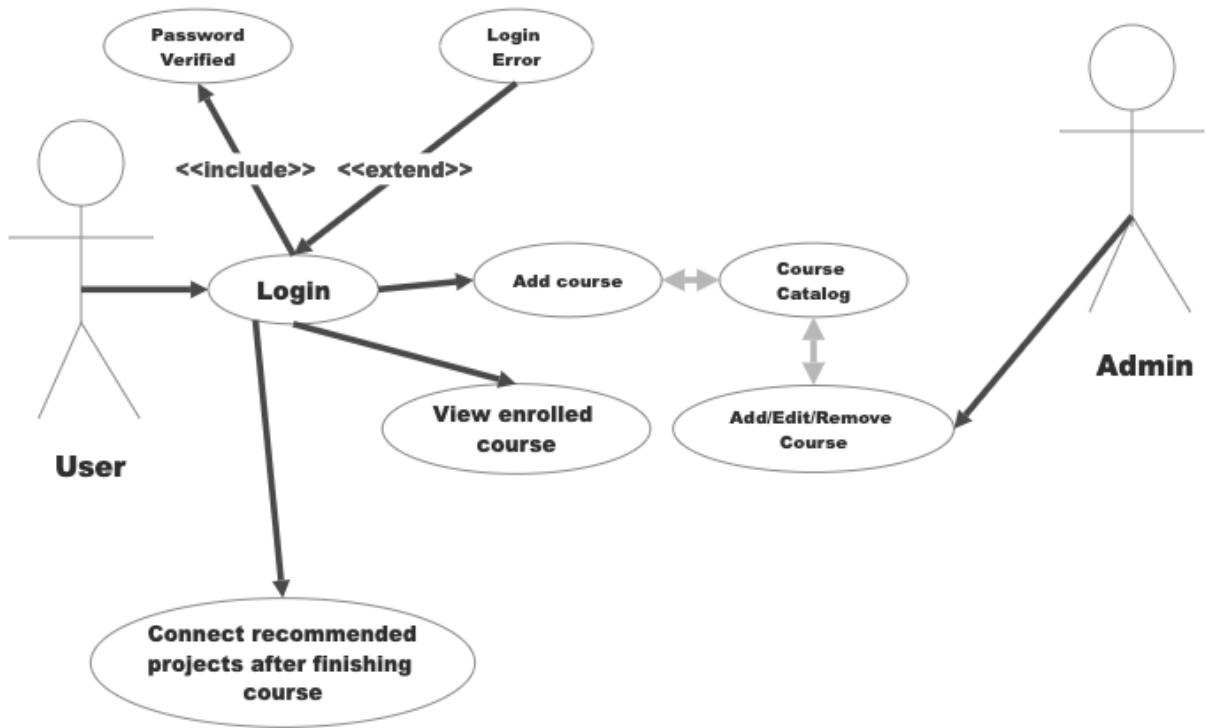
	- Each user will have an individualized Dashboard UI as their learning progress differs from everyone else's
--	--

Use Case Diagrams

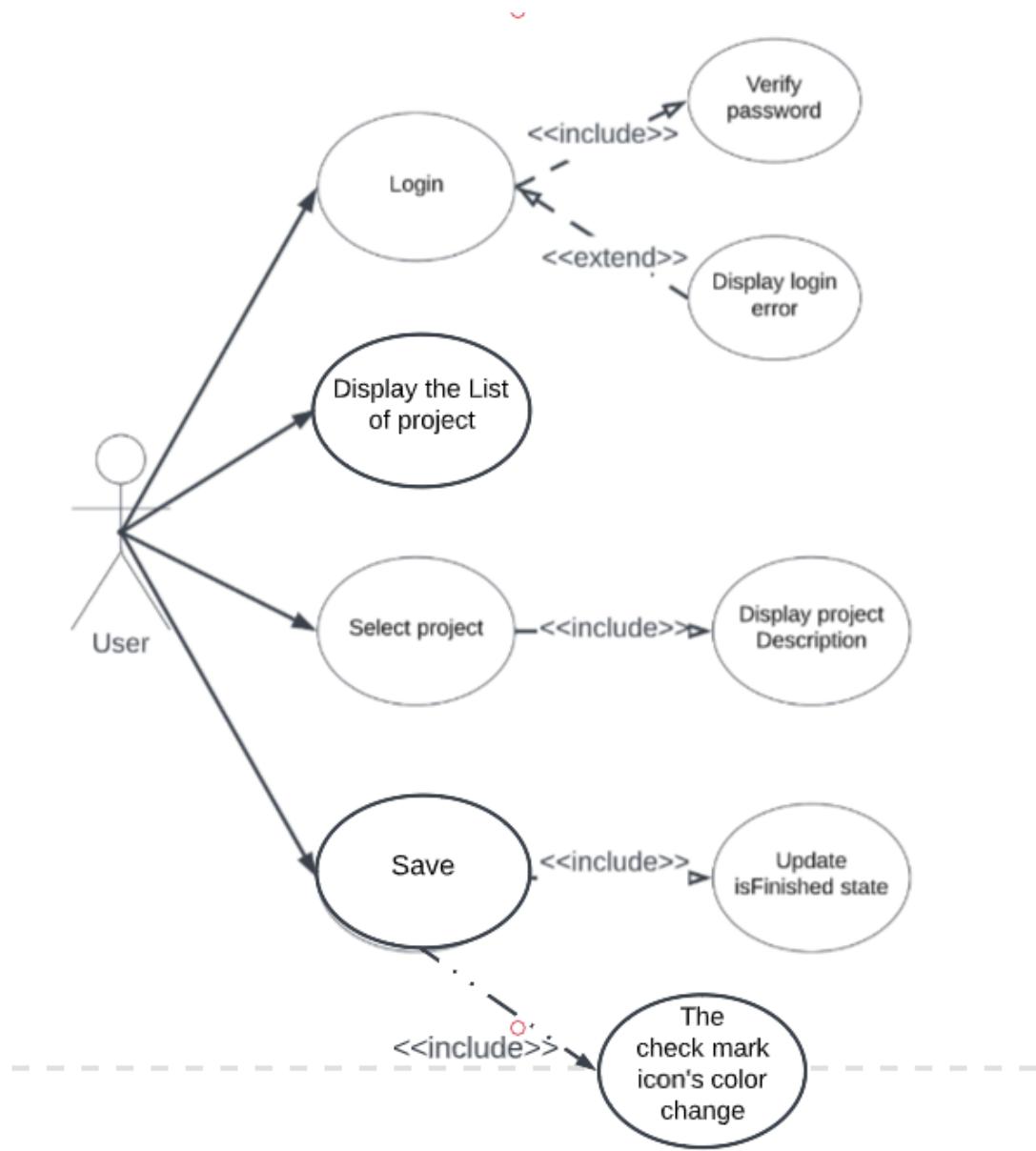
Feature #1 Interview Questions (Core Requirement C1 and C3)



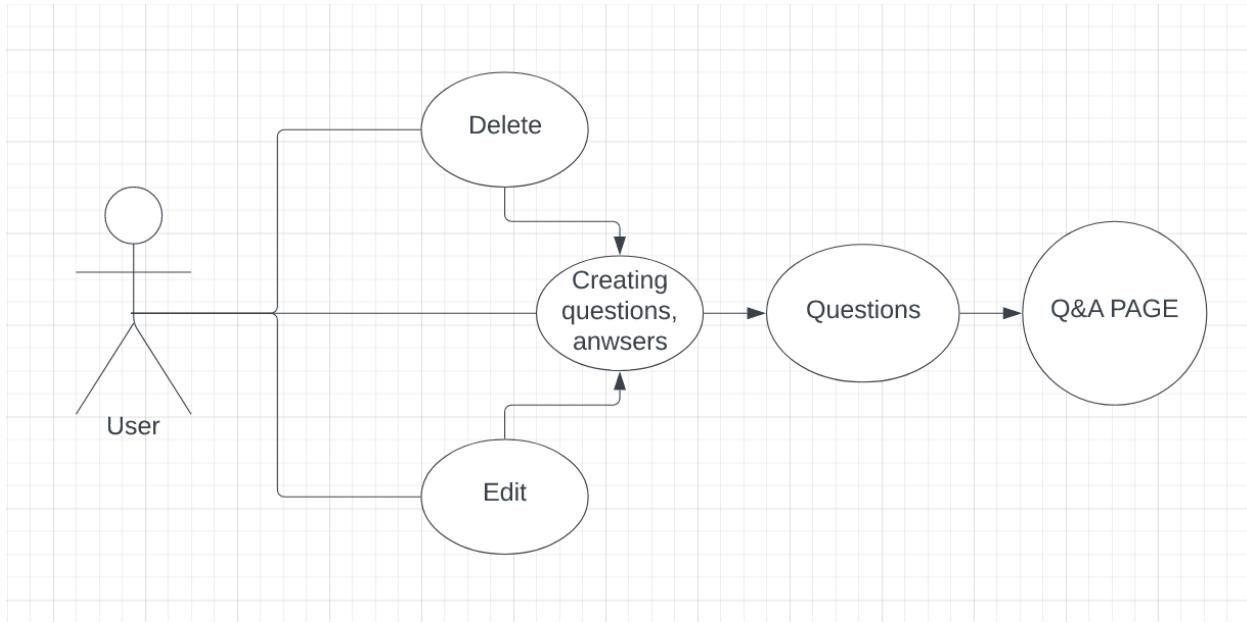
Feature #2 Learning Hub (Core Requirement C1, C2, C3 and C5)



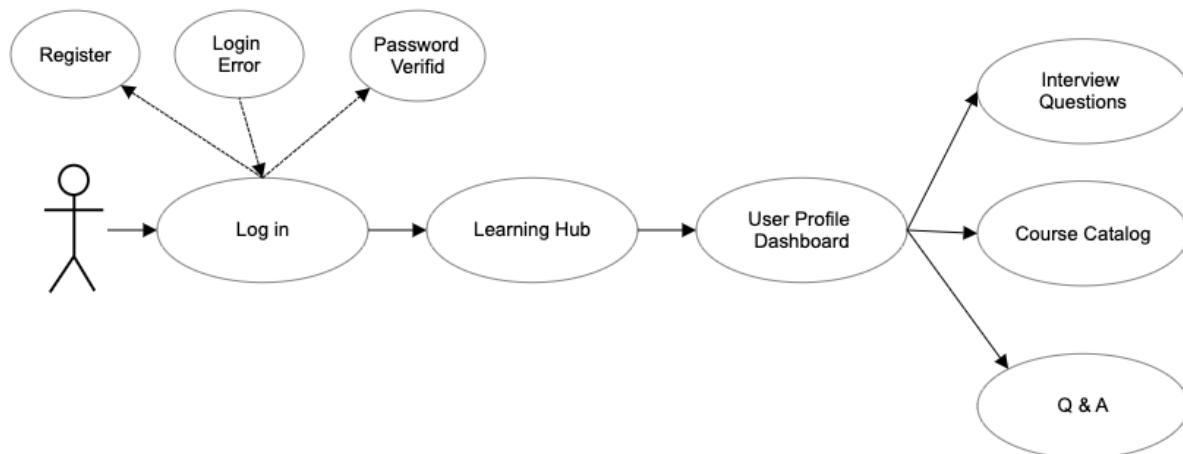
Feature #3 Recommended Projects (Core Requirement C3 & C5)



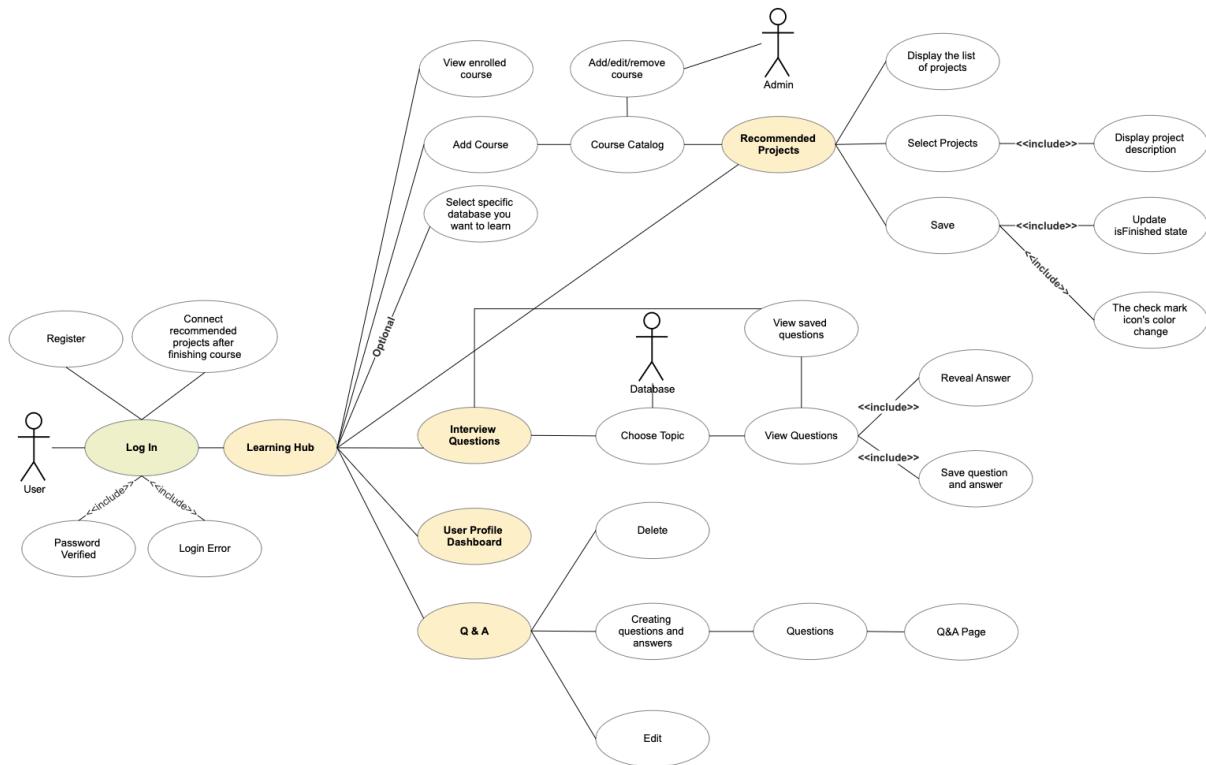
Feature #4 Q&A Page (C3 and C5)



Feature #5 User Profile Dashboard (C5) (Revised)



Overview Use Case

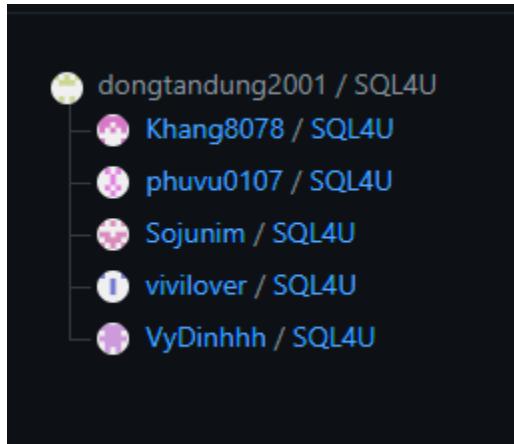


Version control system to collaborate

User Guide

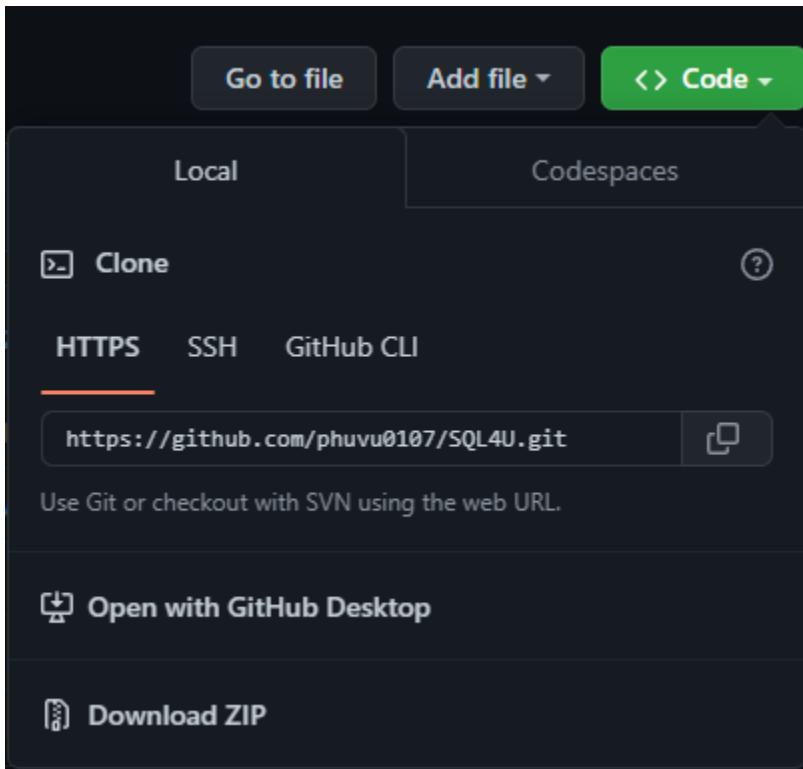
One complete process on Github (or similar)

There would be a main repository created by a member, then others will fork it into their own repository.



Get Started on Forked Repository:

1. git clone <http url of forked github repo>



2. git checkout -b <your branch> ⇒ create new branch to work on
3. git add . ⇒ stash works.
4. git commit -m “<message briefly describe what changed in this commit>” ⇒ commit work.
5. git checkout main ⇒ go back to main branch.
6. git merge <your branch> ⇒ merge your main with latest code.
7. git push ⇒ push code to remote/forked repository.

Code Checkout from Github: Version Control Flow:

1. **Forked Repo: Create pull request after finishing coding and pushing to forked repository:**

About
No description, website, or topics provided.
0 stars
0 forks
5 watch stars

Releases
No releases published
Create a new release

Packages
No packages published
Publish your first package

Languages
JavaScript 88.0%
CSS 11.1%
Other 0.9%

About
No description, website, or topics provided.
0 stars
0 forks
5 watch stars

Releases
No releases published
Create a new release

Packages
No packages published
Publish your first package

Languages
JavaScript 88.0%
CSS 11.1%
Other 0.9%

Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).

Fixed styling, make the page responsive on mobile #34

No description available

[View pull request](#)

Helpful resources
GitHub Community Guidelines

Leave a comment

Attach files by dragging & dropping, selecting or pasting them.

Allow edits by maintainers

[Create pull request](#)

Remember, contributions to this repository should follow our GitHub Community Guidelines.

2 commits 8 files changed 1 contributor

Commits on Dec 2, 2022

Fixed styling, make the page responsive on mobile #35

[Open](#) phuvu0107 wants to merge 2 commits into [dongtandung2001/main](#) from [phuvu0107/main](#)

Conversation 0 Commits 2 Checks 0 Files changed 8

phuvu0107 commented now

No description provided

Truong Phu Vu added 2 commits 7 minutes ago

- Fixed styling, make the page responsive on mobile 923a42f
- Merge branch 'main' of <https://github.com/phuvu0107/SQL4U> 0ccb99b

Add more commits by pushing to the [main](#) branch on [phuvu0107/SQL4U](#).

This branch has no conflicts with the base branch

Only those with [write access](#) to this repository can merge pull requests.

Write Preview

Leave a comment

Attach files by dragging & dropping, selecting or pasting them.

[Close pull request](#) [Comment](#)

Remember, contributions to this repository should follow our GitHub Community Guidelines.

Reviewers
No reviews
Still in progress? Convert to draft

Assignees
No one assigned

Labels
None yet

Projects
None yet

Milestone
No milestone

Development
Successfully merging this pull request may close these issues.
None yet

Notifications
[Unsubscribe](#) Customize
You're receiving notifications because you authored the thread.

2. Owner of repository: Review and merge pull request:

Label issues and pull requests for new contributors

Now, GitHub will help potential first-time contributors [discover issues](#) labeled with [good first issue](#).

Dismiss

Filters Labels 9 Milestones 0 New pull request

1 Open 34 Closed

Fixed styling, make the page responsive on mobile #35 opened 9 hours ago by phuvu0107

Author ▾ Label ▾ Projects ▾ Milestones ▾ Reviews ▾ Assignee ▾ Sort ▾

Fixed styling, make the page responsive on mobile #35

The screenshot shows a GitHub pull request interface. At the top, a green button says "Open" and a message states "phuvu0107 wants to merge 2 commits into `dongtandung2001:main` from `phuvu0107:main`". Below this are tabs for "Conversation" (0), "Commits" (2), "Checks" (0), and "Files changed" (8). The commit list shows:

- phuvu0107 commented 9 hours ago: "No description provided."
- Truong Phu Vu added 2 commits 9 hours ago:
 - Fixed styling, make the page responsive on mobile (commit hash: 923a42f)
 - Merge branch 'main' of <https://github.com/phuvu0107/SQL4U> (commit hash: 0ccb990)

Below the commits, a note says "Add more commits by pushing to the `main` branch on [phuvu0107/SQL4U](#)".

In the middle section, there are two items under "Branch protection rules":

- "Require approval from specific reviewers before merging" (with a link to "Branch protection rules")
- "This branch has no conflicts with the base branch" (with a note that merging can be performed automatically)

A green "Merge pull request" button is at the bottom, along with a note: "You can also open this in GitHub Desktop or view command line instructions".

The bottom section shows the merged state of the pull request:

Fixed styling, make the page responsive on mobile #35

Merged dongtandung2001 merged 2 commits into `dongtandung2001:main` from `phuvu0107:main` now

3. Pull latest code to local repository:

`git pull`

`git pull --rebase` ⇒ use this if there is something new committed on your local

repository.

```
frontend/src/component/footer/footerCSS.css      |  1 +
frontend/src/component/interview/QuestionList.jsx | 47 +-----
frontend/src/component/interview/interview.jsx    |  8 ++--
.../src/component/learningHub/CourseCatalog.jsx   |  2 ++
frontend/src/component/learningHub/learningHub.css| 25 ++++++++
frontend/src/component/qna/TopicList.jsx          |  4 ++
8 files changed, 56 insertions(+), 61 deletions(-)
```

Dung@DESKTOP-I6P06CE MINGW64 /e/SQL4U (main)

\$

4. Checkout to your working branch to not mess up with main branch:

```
Dung@DESKTOP-I6P06CE MINGW64 /e/SQL4U (main)
$ git checkout frontend
Switched to branch 'frontend'
```

Dung@DESKTOP-I6P06CE MINGW64 /e/SQL4U (frontend)

\$

Add -b flag if the branch does not exist yet.

5. Merge latest code to your branch:

```
Dung@DESKTOP-I6P06CE MINGW64 /e/SQL4U (frontend)
$ git merge main
Updating 9de066d..153ba93
Fast-forward
 backend/routes/question.js           | 16 +-
 frontend/package-lock.json          | 105 ++++++-----
 frontend/package.json                |  4 ++
 frontend/public/index.html           |  2 ++
 frontend/src/App.css                 | 60 ++++++-
 frontend/src/App.js                  | 98 ++++++-----
 frontend/src/component/common/input.jsx|  6 +-
 frontend/src/component/footer/footer.jsx| 62 ++++++
 frontend/src/component/footer/footerCSS.css| 18 ++
 frontend/src/component/interview/QuestionList.jsx| 63 +-----+
 frontend/src/component/interview/interview.css| 56 +++++-
 frontend/src/component/interview/interview.jsx| 57 +++++-
 .../src/component/learningHub/CourseCatalog.jsx|  6 +-
 .../src/component/learningHub/IndividualCourse.jsx|  3 ++
 .../src/component/learningHub/TutorialForm.jsx|  1 -
 .../src/component/learningHub/TutorialPage.jsx|  4 ++
 frontend/src/component/learningHub/learningHub.css| 120 ++++++-----
 frontend/src/component/login/login.jsx|  2 ++
 frontend/src/component/login/loginCSS.css|  1 +
 frontend/src/component/navbar/navbar.jsx|  6 ++
 .../src/component/project/IndividualProject.jsx| 34 +++
 frontend/src/component/project/ProjectPage.jsx| 109 ++++++-----
 frontend/src/component/project/projects.js| 187 -----
 .../src/component/project/recommendedProjects.css| 11 ++
 frontend/src/component/qna/AnswerList.jsx| 49 +---+
 frontend/src/component/qna/CreatePost.jsx|  4 +-
 frontend/src/component/qna/Question.jsx| 39 +---+
 frontend/src/component/qna/QuestionAndAnswer.jsx| 129 ++++++-----
 frontend/src/component/qna/TopicList.jsx| 69 +---+
 frontend/src/component/qna/qna-style.css|  2 ++
 frontend/src/component/register/registerCSS.css|  1 +
 frontend/src/services/questionService.js| 51 +---+
32 files changed, 793 insertions(+), 582 deletions(-)
create mode 100644 frontend/src/component/footer/footer.jsx
create mode 100644 frontend/src/component/footer/footerCSS.css
delete mode 100644 frontend/src/component/project/projects.js
```

Dung@DESKTOP-I6P06CE MINGW64 /e/SQL4U (frontend)

\$

6. Commit files after finish coding your part:

```
Dung@DESKTOP-I6P06CE MINGW64 /e/SQL4U (frontend)
$ git add .

Dung@DESKTOP-I6P06CE MINGW64 /e/SQL4U (frontend)
$ git commit -m "Example commit for report"
[frontend b2c51da] Example commit for report
 1 file changed, 1 insertion(+), 1 deletion(-)
```

7. Checkout back to main then merge with your branch.

```
Dung@DESKTOP-I6P06CE MINGW64 /e/SQL4U (frontend)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

Dung@DESKTOP-I6P06CE MINGW64 /e/SQL4U (main)
$ git merge frontend
Updating 153ba93..b2c51da
Fast-forward
 frontend/.env.development | 2 ++
 1 file changed, 1 insertion(+), 1 deletion(-)
```

8. Push back to the remote repository

```
Dung@DESKTOP-I6P06CE MINGW64 /e/SQL4U (main)
$ git push
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 366 bytes | 366.00 KiB/s, done.
Total 4 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/dongtandung2001/SQL4U.git
 153ba93..b2c51da  main -> main
```

Note: Check if there is anything new on the main remote repository, if yes, sync the main repo with your forked repo do this step instead:

git pull --rebase

git push

9. Other Forked Repo: Sync with main Repo

The screenshot shows a GitHub repository page for 'phuvu0107 / SQL4U'. The 'Code' tab is selected, showing the 'main' branch. A tooltip indicates that the 'main' branch is 2 commits behind the upstream 'main' branch. The repository has 2 branches and 0 tags. The 'About' section notes that the repository has no description, website, or topics provided. It shows 0 stars, 0 watching, and 5 forks. The 'Languages' section shows a chart where JavaScript accounts for 88.0%, CSS for 11.1%, and Other for 0.9%. The 'Sync fork' button is highlighted.

10. Repeat from step 1.

Additional things (not mentioned above)

Design Document

1. All endpoints:

```
<Routes>
  <Route element={<PrivateRoutes />}>
    <Route element={<Dashboard />} path='/' />

    <Route path='/interview' element={<InterviewQuestion />} />
    <Route path='/interview/:id' element={<InterviewQuestionForm />} />

    <Route path='/catalog' element={<CoursesCard />} />
    <Route
      path='/catalog/:courseId/project'
      element={<RecommendProject />}
    />
    <Route
      path='/catalog/:courseId/project/:projectId'
      element={<ProjectPage />}
    />
    <Route
      path='/catalog/:courseId/project/add/:id'
      element={<ProjectForm />}
    />
    <Route
      path='/catalog/:courseId/tutorial/add/:id'
      element={<TutorialForm />}
    />
    <Route path='/catalog/addOrEdit/:id' element={<CourseForm />} />
    <Route path='/catalog/: courseId' element={<IndividualCourse />} />
    <Route
      path='/catalog/:courseId/tutorial/:tutorialId'
      element={<TutorialPage />}
    />

    <Route path='/qna' element={<QnA />} />
    <Route path='/qna/:id' element={<QuestionAndAnswer />} />
    <Route path='/qna/edit/:id' element={<CreatePost />} />

    <Route path='/hub' element={<LearningHub />} />
  </Route>
  {/* <Route path='/' element={<Dashboard />} /> */}
  <Route path='/about' element={<About />} />
  <Route path='/contact' element={<Contact />} />

  <Route path='/logout' element={<Logout />} />
  <Route path='/profile' element={<Profile />} />

  <Route path='/login' element={<Login />} />
  <Route path='/register' element={<Register />} />
  <Route path='/not-found' element={<NotFound />} />

  <Route path='/' element={<Navigate to={"/"} />} />
  <Route path='*' element={<Navigate to={"not-found"} />} />
</Routes>
```

Learning hub:

1. Main page: <https://www.sql4u.dev/catalog>
2. Specific course:
<https://www.sql4u.dev/catalog/6385b7a9870e807b4cfdccd2>
3. Specific Tutorial of that course:
<https://www.sql4u.dev/catalog/6385b7a9870e807b4cfdccd2/tutorial/6385d166870e807b4cfdd100>
4. Add new Course (admin): <https://www.sql4u.dev/catalog/addOrEdit/new>
5. Edit course (admin):
<https://www.sql4u.dev/catalog/addOrEdit/6385b7a9870e807b4cfdccd2>
6. Add new tutorial for a course (admin):
<https://www.sql4u.dev/catalog/6385b7a9870e807b4cfdccd2/tutorial/add/new>

7. Edit a tutorial of a course(admin):

<https://www.sql4u.dev/catalog/6385b7a9870e807b4cfdccd2/tutorial/add/6385d166870e807b4cfdd100>

Interview Question:

1. Main page: <https://www.sql4u.dev/interview>
2. Add new question (admin): <https://www.sql4u.dev/interview/new>
3. Edit question(admin):
<https://www.sql4u.dev/catalog/6385b7a9870e807b4cfdccd2/tutorial/add/6385d166870e807b4cfdd100>

Recommend Project:

1. Project list of a course:
<https://www.sql4u.dev/catalog/6385b7a9870e807b4cfdccd2/project>
2. Project Detail:
<https://www.sql4u.dev/catalog/6385b7a9870e807b4cfdccd2/project/6388655eeec79bb533e8ad1e>
3. Add new Project for a course (admin):
<https://www.sql4u.dev/catalog/6385b7a9870e807b4cfdccd2/project/add/new>
4. Edit a Project Detail (admin):
<https://www.sql4u.dev/catalog/6385b7a9870e807b4cfdccd2/project/add/638865eeec79bb533e8ad1e>

QnA:

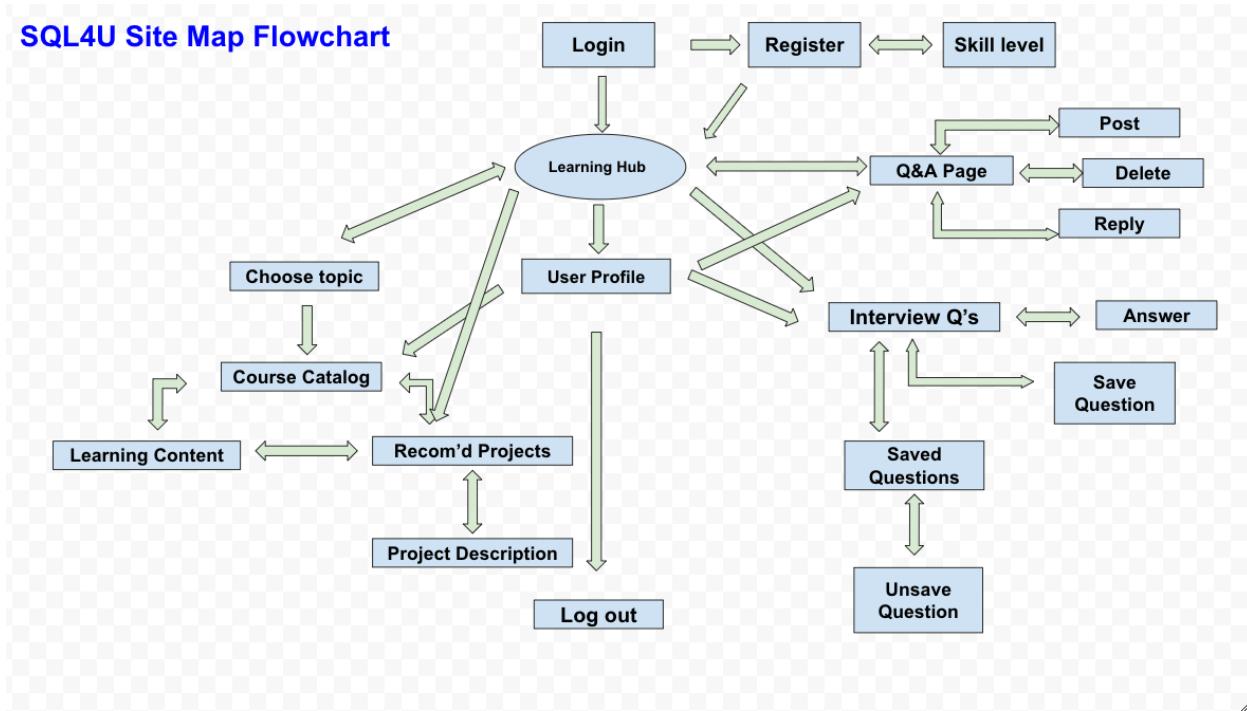
1. Main page: <https://www.sql4u.dev/qna>
2. Specific question: <https://www.sql4u.dev/qna/638481fe870e807b4cfdc510>
3. Edit question (only user can edit their question):
<https://www.sql4u.dev/qna/edit/6387e775eec79bb533e896f3>

Explain why each of your suggested features is innovative and will be useful from the user's perspective.

- Learning Hub

- Learning Hub is unique and useful in the sense that it provides users with a vast amount of courses that are relevant to their needs. It also has a filter in case the user decides to focus on a particular topic. This feature also acts as a landing page which leads the user to recommended projects where the user can further their knowledge
- Interview Questions
 - Interview questions provide the user the final prep prior to an interview. This feature is useful since on this page, the user will be able to access a curated list of interview questions that are specific to databases. This feature also has a filter to a more specific topic of the database. The user can also save the questions they liked and can view all of them on the saved questions view.
- Recommended Projects
 - Recommended projects can be accessed through the learning hub or through its own landing page. The user will need to choose a topic and they'll be able to access the project related to that topic. Recommended projects also has a progress check mark to show the user whether the project has been completed or not. Projects are useful since they can be used as a portfolio for interviews and often times people learn better working on a project rather than reading a book as a more hands on approach.
- QnA
 - QnA is where the user can go and ask questions related to the database subject. They can post a question on a specific topic. Other users can also reply to the questions which are listed on the page. QnA page also has a filter where the user can filter the list of questions to find what they are looking for. It is useful because often times people who are learning the subject would have lots of questions so it's nice to have a place where they can ask the community.

Site Map Flowchart



Describe Agile Design Process

We utilized the Agile Design process throughout the project. There were two main approaches:

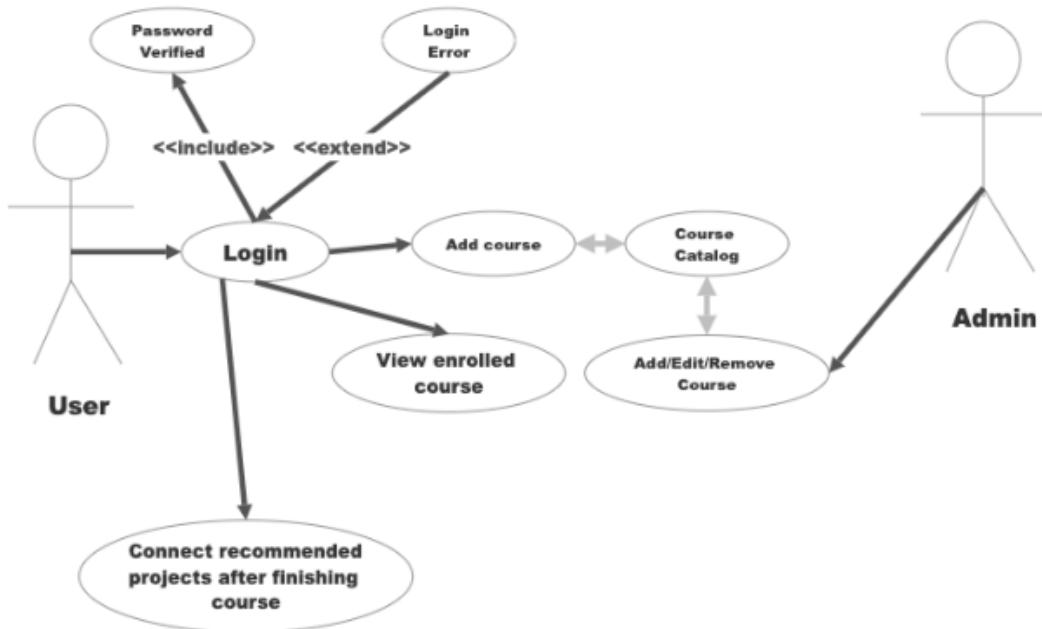
1. The validation from other teams:
 - a. At the second part of each milestone, we've had other teams validating our milestone and offering feedback. We took a look at the report from the validating team and applied necessary changes to our project. Below is how we adopted Agile process by taking feedback from another team into our next milestone.

Feature Summary Table

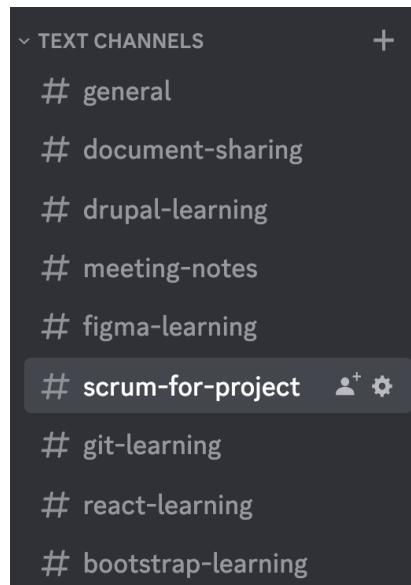
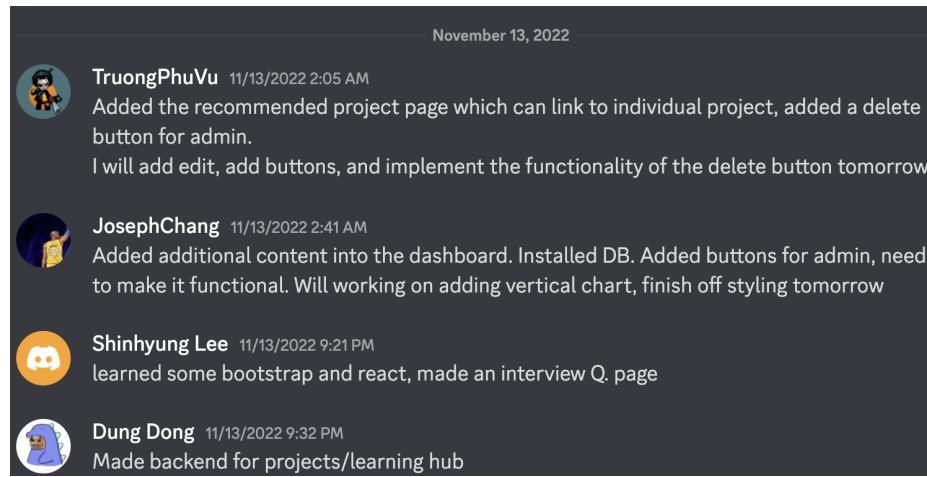
Features	Why is it useful to users	Why is it innovative?	Matching core requirement	Team Composition
1. Technical Blogs	One stop shop for all tech blogs	4 different categorized list of blogs	C2 and C3	Joseph Chang
2. Interview Questions	<p>It helps users know what to expect in an interview.</p> <p>It is also a way to test the knowledge of users.</p>	<p>Categorized topics in DBMS</p> <p>Hidden answers so that users can learn more productively.</p>	C1 and C3	Dung Dong
3. Learning Hub	<p>Provide enough knowledge for <u>beginner</u> about the subject.</p> <p>Build professional background for career success</p>	<p>Improve learning experience for users. Combine useful tutorials in different formats.</p> <p>Easily keep track history of activities and approach problems under each subcategory.</p>	C1, C2, C3 and C5	Khang Huynh
4. Check Progress	<p>Helps users keep track of their progress.</p> <p>Motivates users to continue learning new knowledge.</p> <p>Revision: This feature is no longer a standalone feature. It was incorporated with Learning Hub and Recommended Projects.</p> <p>HOWEVER, it will not be implemented in <u>Milestone2A</u>.</p>	Can be implemented under different features(Learning hub, interview questions)	C5	Shinhyung Lee
5. Recommend Projects	Practice users' newly learned concepts.	Easily know how to apply the concept into real-world problems	C3 and C5	Truong Phu Vu

Use Case ID	C1-3
Use Case Name	Learning Hub
Created By	Khang Huynh
Actor	User
Description and Event Flow	<p>A collection of pages outlining databases, database management systems (DBMS), the distinctions between two common types of databases, and when and why to use each kind of database. Additionally, it will direct users to online lessons and suggested projects while displaying the necessary tools.</p> <ul style="list-style-type: none"> ▪ User logged in and on the homepage ▪ The first page of this feature is "My Course," where the user can see their enrolled courses because each user has an account where all activities are saved. ▪ The first-time user can add a course in the "Course Catalog." ▪ "Course Catalog" contains all courses covering the technical subject. Users can filter courses based on their favorite topic. ▪ The first-time user (level: beginner) can start by learning the basic concepts of database management, such as databases, transactions, relational models, normalization techniques, tables, different keys in schemas, etc. ▪ The first-time user (level: professional) can skip the basic concepts and select their favorite topic/courses. ▪ Each course shows a list of tutorials. When users complete all tutorials of this course, they can see a button that leads them to Recommended Projects. Based on what they learned, it will show some projects and help users apply practical knowledge of the database

Learning Hub (updated)



- b. Second Agile process we went through was within our team. We created a channel in Discord specific to SCRUM where we would update each members on what we've done, what we are planning to do, any challenges we are facing and how to solve them. The continuous open communication helped us build an Agile process where we can bounce ideas off each other in order to keep improving the project. We've also had numerous meetings for more intense obstacles and/or needed a bigger alignment across the team. After the team meeting, we would share the notes and responsibilities that were handed out to individuals to enforce our alignment which led to the success of our project.



November 9, 2022



Shinyung Lee 11/09/2022 11:43 PM

Nov 9 Meeting Note

In preparation for:

- Nov 15 (4:30pm), Website Barebone Presentation
- Nov 16 (11pm), Milestone 2A Report on Canvas

Figma

<https://www.figma.com/file/OorFnv8sIHx9VBxeOazZik/SQL4U?node-id=0%3A1&t=ub3jxTxiVmsWPOyX-0>

- Mostly looks good, we can come back to the details after milestone2A. For now, focus on the 4 features for the presentation.
- Features to give more focus on after Milestone2A: Advising Quiz, Q&A
- Joseph mentioned we can work on choosing the colors when we code

*Advising Quiz: placement test

*Q & A: we will have a) frequently asked questions for each topic AND b) user created Q&A questions

Individual Homework: work on HTML & CSS

Dung's skeleton repo:

<https://github.com/dongtandung2001/SQL4U>

**Caution: after you fork & clone the