I3301 – I3350 – I3302

CodeQuest

17547, Nasr Ibrahim

Summary

Contents

1. Introduction

In modern days, technology is evolving very fast, even faster than our brains can comprehend or more importantly keep up. With this evolution comes he increased demand for jobs within this domain. When the internet was first developed, only a handful of people were needed to see what the hype was about. It was not until the true potential was discovered that any company had a full specialist team dedicated solely for this purpose. Moreover, as time moved on, new vulnerabilities were discovered. Same as before, what started as one or two security people turned into a group of cybersecurity specialists.

Also in modern society, what used to take people months to comprehend is now taking children and babies days to understand. Of course, we are talking about electronic devices like smart phones, computers, and TVs. Tech enthusiasts considered the idea behind a touchable screen a miracle. A month-old will take one look and start swiping and going to YouTube. “What are the implications?” is an important question and topic regarding this phenomenon in health and morals. However, it is not asked enough concerning their future careers or even thought process in this domain. How they are being introduced to technology will influence how they perceive it in the long run.

These specialists, whether in web development, front-end, back-end, software, cybersecurity…etc., are in essence students. They finish high school, pick a major, go to university, and after graduating, start working. Due to the increased demand of such jobs, there needs to be an increased supply in the workforce. This workforce did not start development in university. It started when the highschoolers started considering the major because of the opportunities. This is our target audience.

There is an app called “Duo Lingo”. It is specialized in teaching languages to users. What fascinated me when I stared using it is the simplicity of using by laying out the correct path. This helped me to focus on the steps to reach the mastery of Spanish and cut off distractions. This facility was not provided when I started learning coding since when I started HTML, I found something called CSS, and before finishing what I started I started discovering CSS. Then it happened with JavaScript and before I knew it, I was burned out from learning multiple languages at once without a correct base and having false expectations. From this, I got the idea for “CodeQuest”. Its purpose is to slowly introduce the user to a certain language and when they pass a certain test, they go on to the other. This way, potential computer scientists would be setting on the correct path to have a strong base in coding, while being motivated to unlock new heights and skills.

2. Analysis

2.1. Business Rules

Like any application with users, each user has a unique profile that they need to access the application. Each user will have a unique automatically generated userID, username and password, name, and phone number (which would be used for any two-step verification process).

Lessons will have unique lessonID and lessonTitle. In addition, each one will have sets of purposes and objectives, which will explain to the user the importance of the language they want to learn before starting.

The application will have an online aspect and an offline one.

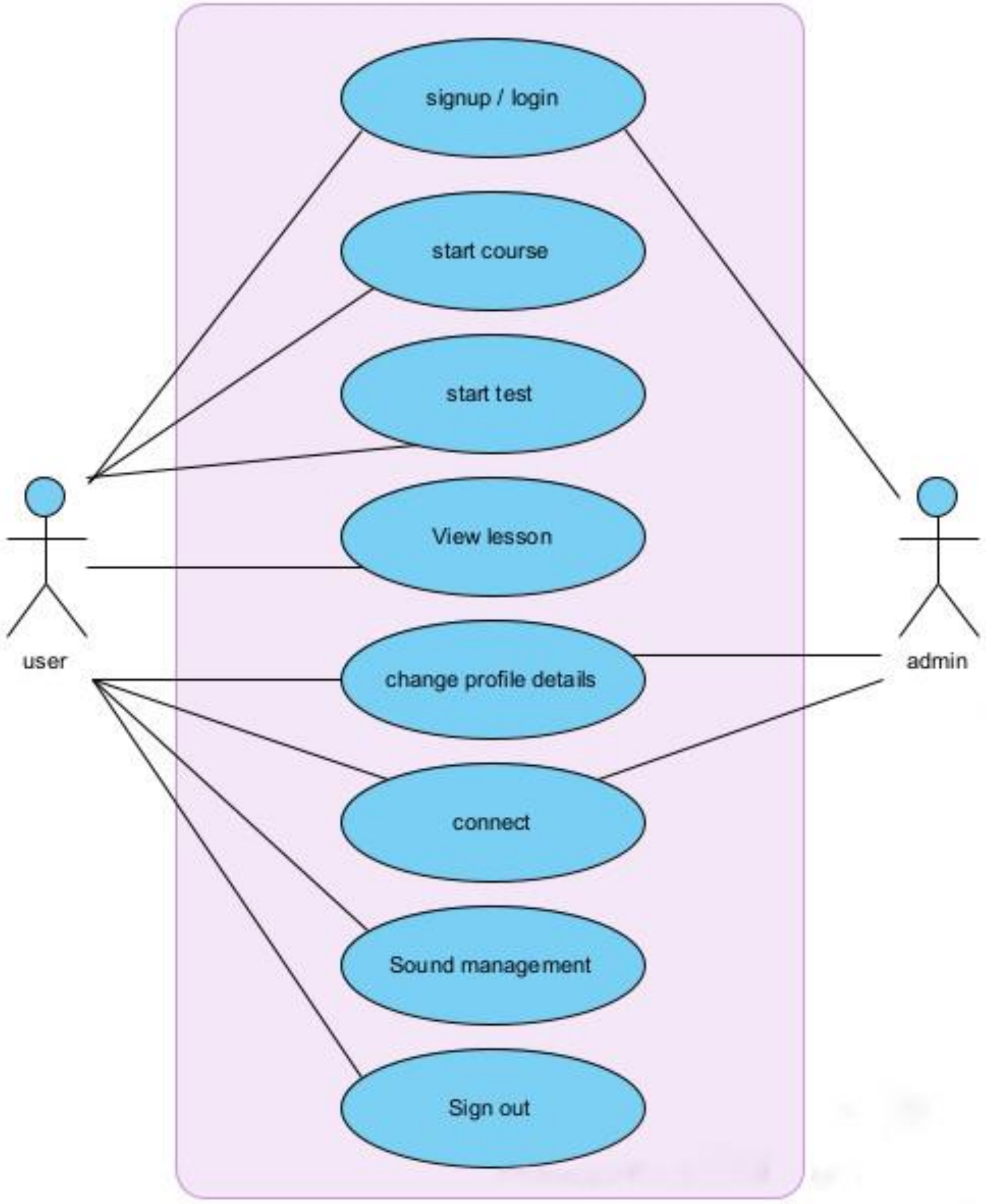
The online database works based on WampServer with php code. This will be used to verify that the correct username and password are used on Log In and then will get all needed information regarding the user profile and respective lessons. In addition, the user will have the ability to change some information, and these changes will be implemented across everything. This will be governed by the admin or the user manager, which will verify changes before implementation. There might also be a list to connect users together.

In addition, it will contain a table of all lessons. The trick is connecting the two tables, user and lesson. There will be two tables showing the relations between them. One will show the lessons a user is taking, the other will show unlocked lessons.

On the other hand, the offline preferences settings will store the information received from the online database regarding the specific user profile and any information regarding the personal device settings. This will not be accessed by the admin since that role will finish upon sending the needed information.

In addition, in an offline internal text, there will be all details regarding the lesson. This will be objectives and purposes, text course, and multiple-choice questions. Based on the java code, only the needed information will be extracted.

2.2. Use case



1. User Signs Up / Logs in

The user initiates the sign-up process by providing necessary details or log in by entering valid credentials. The admin validates the information.

2. User starts course

The user can start any course they want after seeing the purposes and objectives of the coding language.

3. User starts test

The user can start a test to measure progress gained

4. User Views Lesson

The user selects a lesson from the Lesson Menu, and the LessonCourse screen is displayed. The user can navigate through the lesson pages.

5. User Changes Profile Details

The user accesses the profile settings and modifies details such as username, email, or password. The admin validates any changes.

6. User connects to other users

The user can view and connect to others using the application. The admin keeps track of followers list.

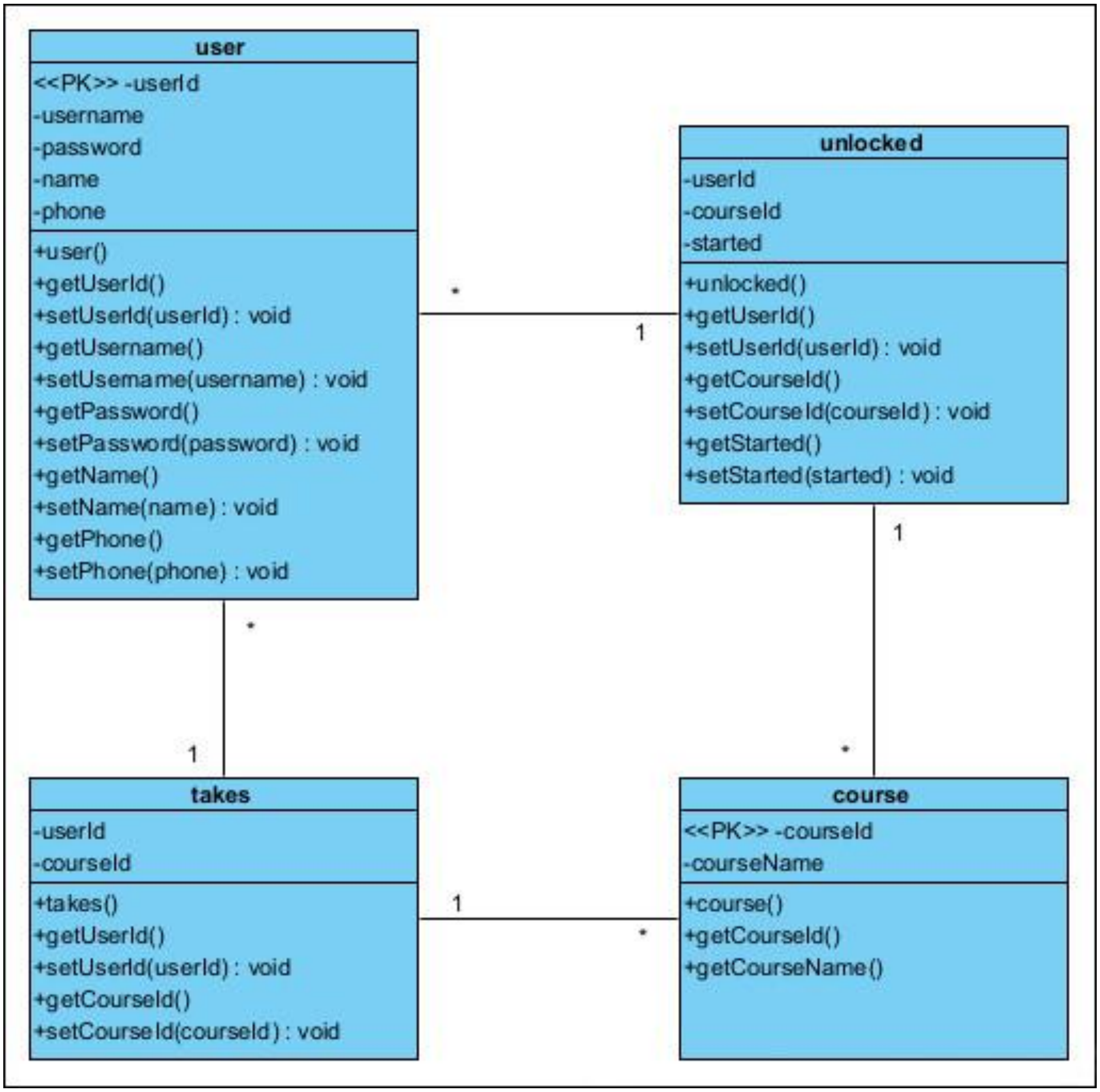
7. User Adjusts Sound Settings

The user adjusts sound-related settings by toggling the sound switch to turn on or off sound effects.

8. User Signs Out

The user initiates the sign-out process by clicking the "Sign Out" button. This action clears user data and returns the user to the login screen.

2.3. Class diagram



For the classes, we have as primary classes: user and course.
user:
Has as attributes unique userId (PK), username, password, name, and phone number. All these have setters and getters to facilitate their changing.
course:
Has as attributes a unique courseId (PK) and courseName. Only getters since each course will already be set with these attributes.

In addition to these two, we have “takes” and “unlocked”.
takes:
It means “user” takes “course”. It has as attributes userId and courseId both FKs.

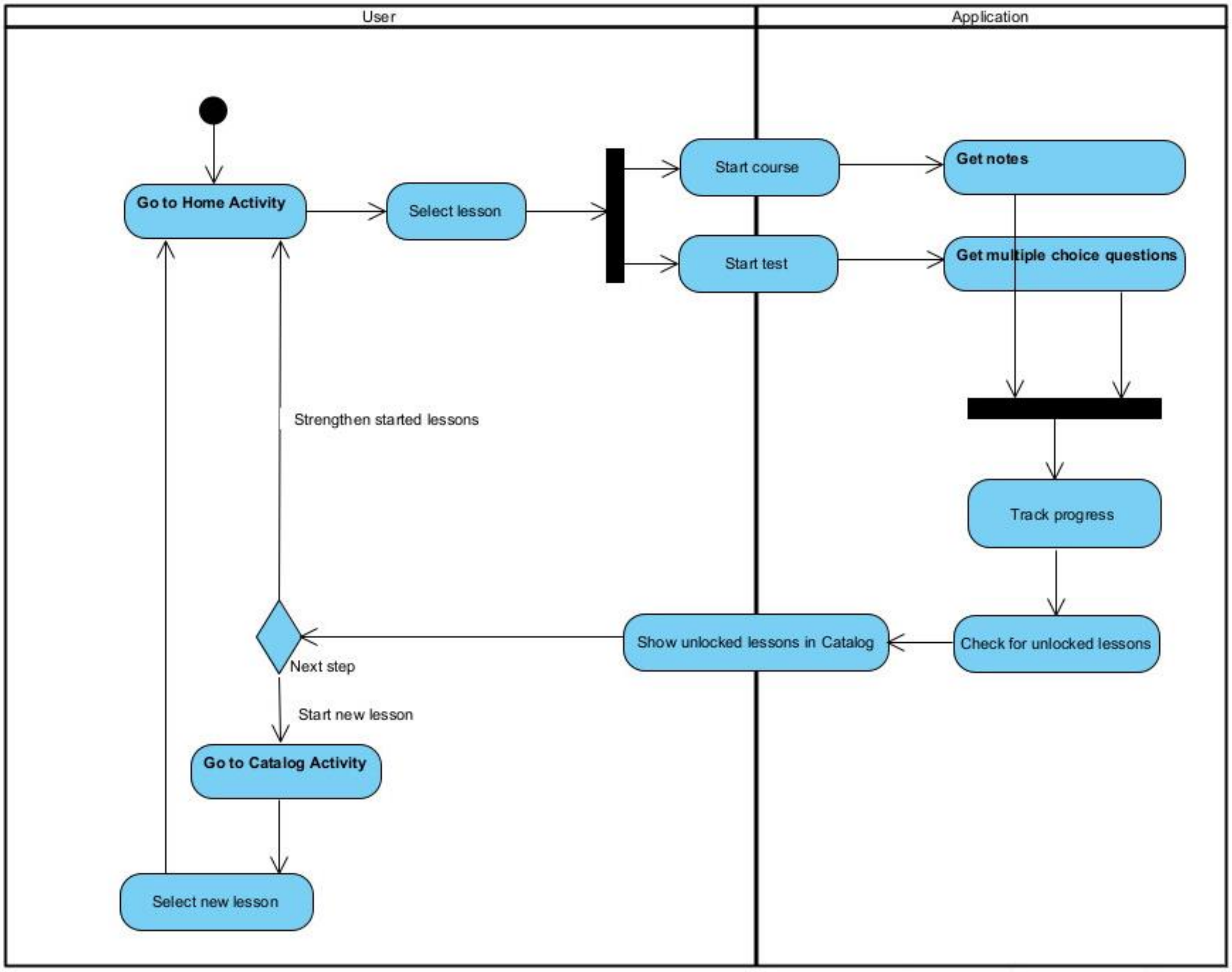
The relation between user and course is many-to-many since multiple users can have multiple courses. The "takes" class then has one-to-many with each since there exists one record of each user taking each course.

unlocked:

It means "user" unlocked "course". It has as attributes userId and courseId both FKs and boolean started to show if the user started the course or not.

Same as before, the relation between user and course is many-to-many. Then "unlocked" class has one-to-many with each.

2.4. Activity Diagram



The above is the main activity diagram which governs the main idea behind the application.

A user would start by going to the Home activity. There they would a certain lesson.

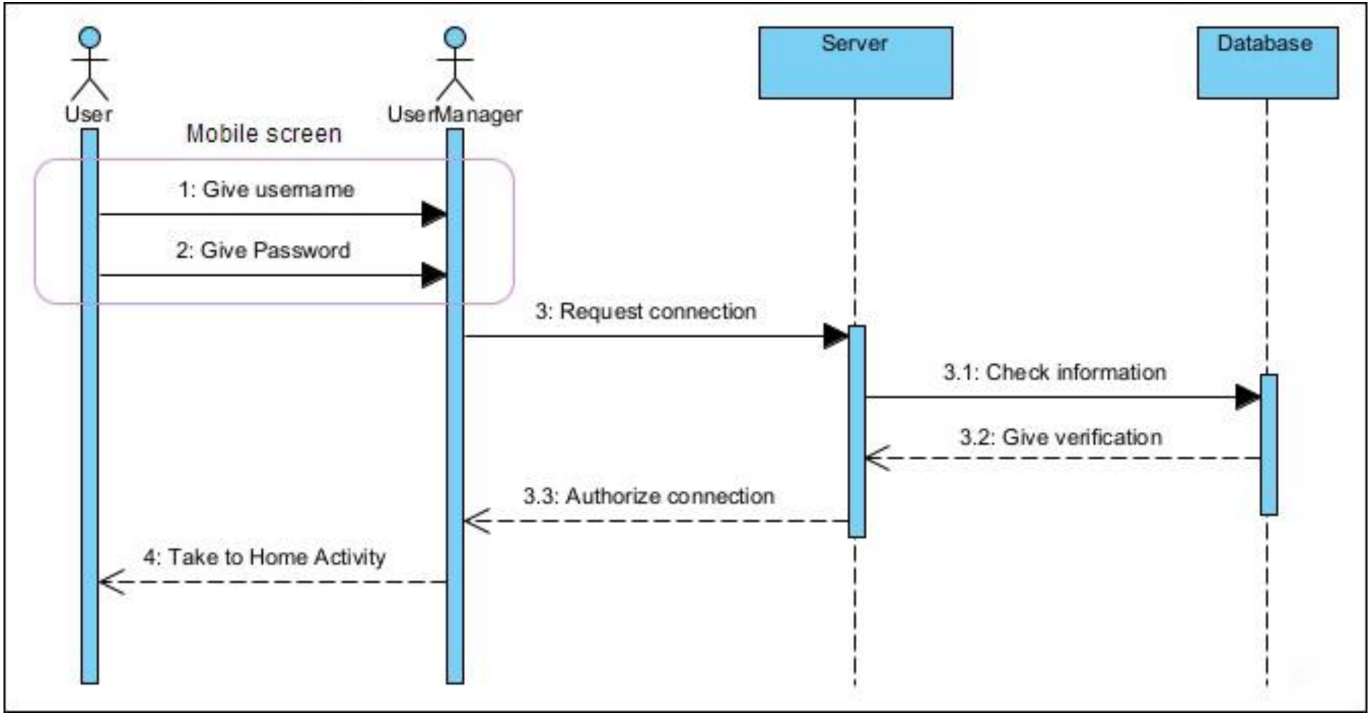
They would have two choices: either start a course or a test. If they choose a course, the application would get notes. If test, it would get the multiple choice questions. In both cases it would track progress and would check for unlocked courses.

These unlocked courses would be shown in the catalog.

Here the user has to take a decision. Either continue strengthening their skills in already started lessons, therefore go directly to the Home activity/ Or start a new lesson. Then they would go to the Catalog activity. There they would choose a new lesson, and then go back ot the Home activity to start the learning process again.

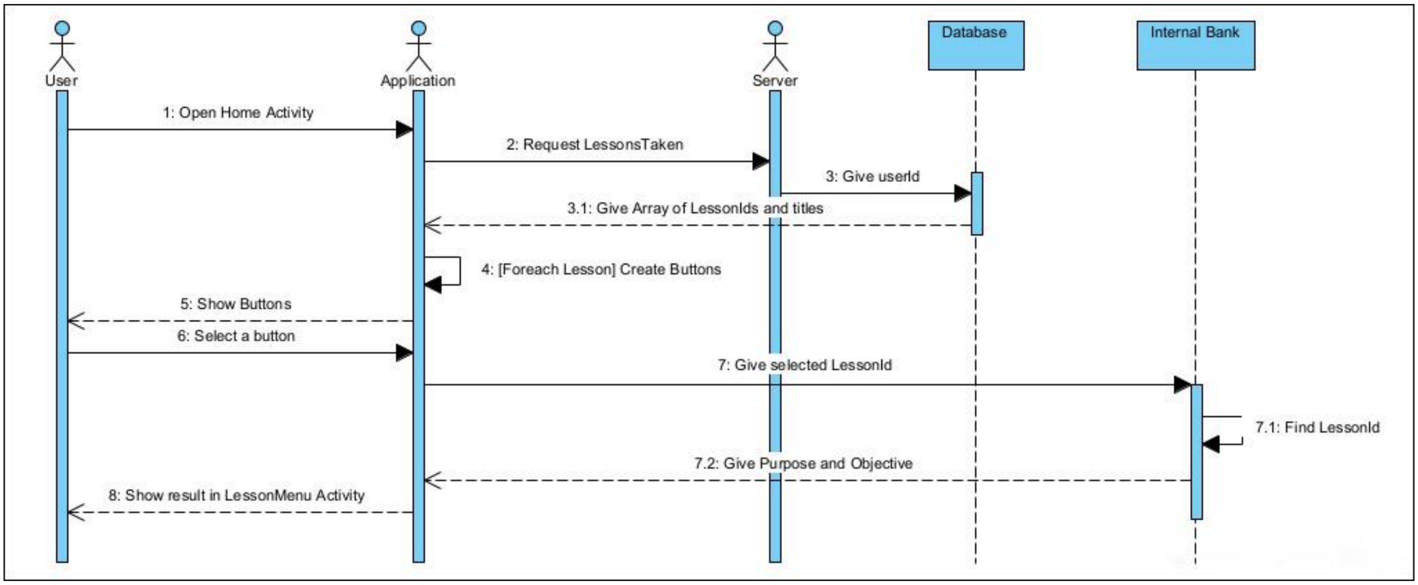
Notice that there is no final node. That's because the process is continuous. There is no point where a lesson would become inaccessible or a user would finish and stop.

2.5. Sequence Diagram



This first sequence diagram is the simple user log in.

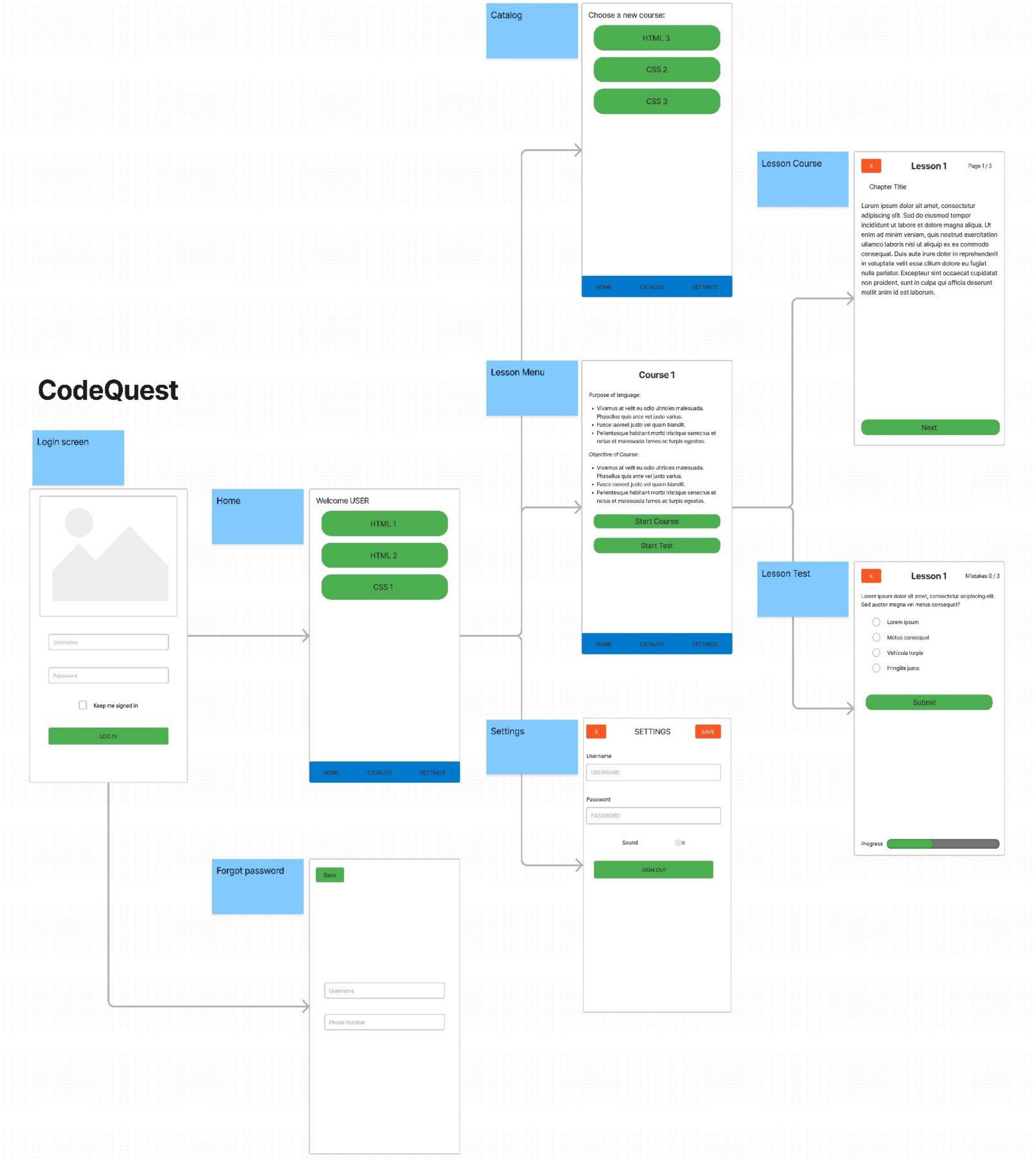
1. The user first gives the UserManager the username and password. This is bounded by the mobile screen.
2. The user manager would request the connection from the server.
3. In turn the server would check the information in the database, which would return a verification if correct or not.
4. Based on this, the server would authorize a connection or not. If valid, the UserManager would take the user to the Home activity.


User
Application
Server
Database
Internal Bank
1: Open Home Activity
2: Request LessonsTaken
3: Give userId
3.1: Give Array of LessonIds and titles
4: [Foreach Lesson] Create Buttons
5: Show Buttons
6: Select a button
7: Give selected LessonId
7.1: Find LessonId
7.2: Give Purpose and Objective
8: Show result in LessonMenu Activity

This sequence represents how the lesson buttons would appear in the Home activity and what happens when the user selects one.

1. When the user first opens the Home activity, the application would request the LessonsTaken from the server.
2. The server would give the userId to the database, which would return an array of lessonIds and titles to the application.
3. The application would loop through the array and create a button for each. Then it would show the buttons to the user.
4. Now that the user sees the button they would choose one. To get the purposes and objectives, the application would send the lessonId to the internal bank.
5. The bank would search for the lessonId. When it finds it, it would send the respective information to the application which in turn would show the user the result.

2.6. Screens description



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Purpose of language:

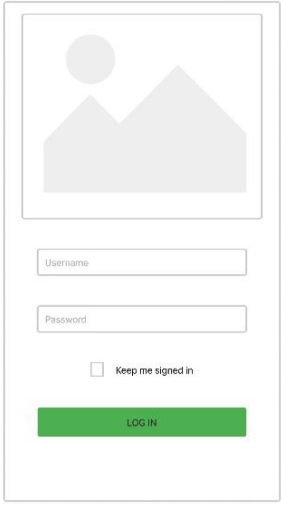
- Vivamus at velit eu odio ultricies malesuada. Phasellus quis ante vel justo varius.
- Fusce laoreet justo vel quam blandit.
- Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas.

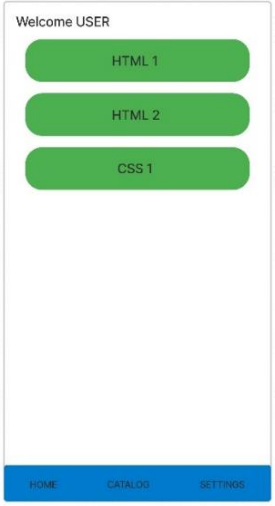
Objective of Course:

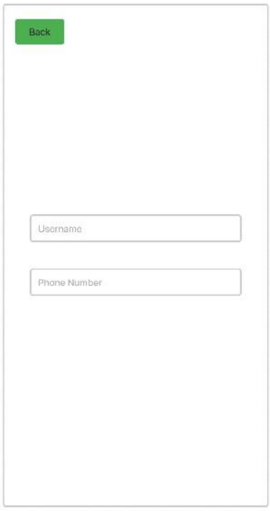
- Vivamus at velit eu odio ultricies malesuada. Phasellus quis ante vel justo varius.
- Fusce laoreet justo vel quam blandit.
- Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas.

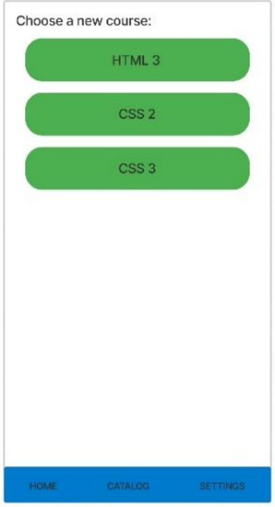
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed auctor magna vel metus consequat?

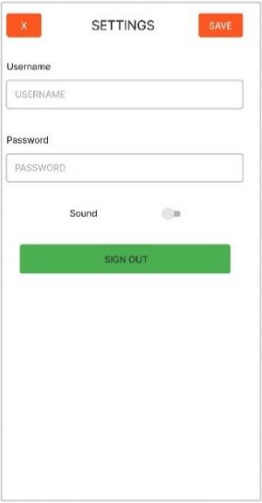
The above is the complete tree plan of my screen sequence in Figma.
The following are each screen shown separately.

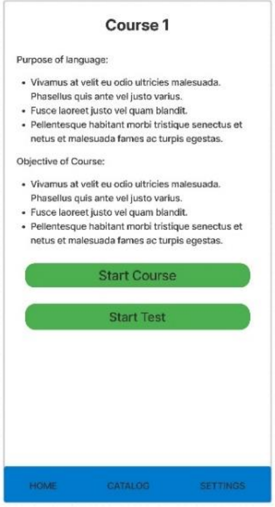












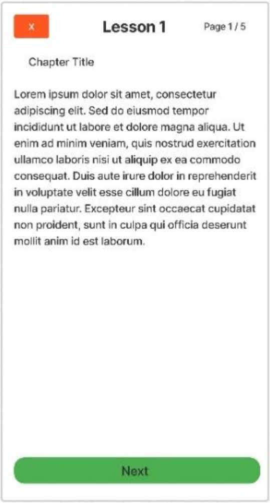
Course 1

Purpose of language:

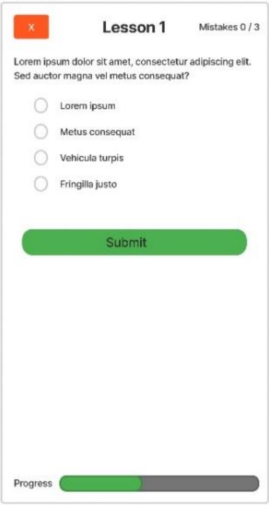
- Vivamus at velit eu odio ultricies malesuada. Phasellus quis ante vel justo varius.
- Fusce laoreet justo vel quam blandit.
- Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas.

Objective of Course:

- Vivamus at velit eu odio ultricies malesuada. Phasellus quis ante vel justo varius.
- Fusce laoreet justo vel quam blandit.
- Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas.



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed auctor magna vel metus consequat?

○ Lorem ipsum
○ Metus consequat
○ Vehicula turpis
○ Fringilla justo

1. Log In:

The Log In screen will be simple. The user will enter their username and password according to the spaces where there will be hints to help. There will also be a “Keep me signed in” button that a user can check to reduce the number of times they log in. In the above image placeholder, we will place the logo.

2. Home:

On entry, the user will be greeted by name in the above text. The lessons they already started can be accessed via buttons in the body where they will be taken to the Lesson Menu screen. In the below navigation bar, they can switch between the Catalog and Settings screens.

3. Forgot Password:

The Forgot Password screen is still being studied. Most probably, the user will enter their username and phone number to change it.

4. Catalog:

The Catalog screen will show lessons that can be started. Of course, the navigation bar below will still be visible.

5. Settings:

In the Settings screen, a user can change their username or password by typing the new ones in place of the old and clicking the Save button. They can also turn the sound on or off by clicking the switch. By clicking the sign out button, they will be redirected to the Log In screen. The navigation bar will not be visible here, instead when the close button is clicked, users will return to the Home screen.

6. Lesson Menu:

In the Lesson Menu screen, the user will first see the Course Title. Under it, they will see the purpose of the language and then the objectives as bullet points. Under these are two buttons, the first will redirect to the Reading Course and the second to the Test. The navigation bar will be visible here.

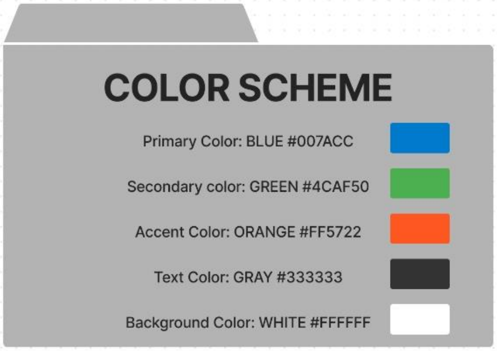
7. Lesson Course:

When a user enters the course, they will see the title, followed by the chapter title and then the text. Under these will be the Next button to switch pages when done reading. In the top-right corner, there will be a text showing the current page out of the total. The navigation bar will not visible and the user will have to confirm exiting when clicking the close button in the top-left corner.

8. Lesson Test:

The test is multiple-choice questions. As usual, above will be the lesson title and the close button. To the right will be a mistake counter, showing the total mistakes out of three (the maximum). In the body, we will start with the question followed by four options, then the submit button. Below will be the progress bar that will fill a bit at each correct answer.

In addition, to the right is the color scheme used.



COLOR SCHEME

Primary Color: BLUE #007ACC	🟦
Secondary color: GREEN #4CAF50	🟩
Accent Color: ORANGE #FF5722	🟧
Text Color: GRAY #333333	⬛
Background Color: WHITE #FFFFFF	⬜

3. Implementation

3.1. Technical environment used

The application was implemented in Android Studio Flamingo 2022.2.1.

- The views mostly consist of the xml layout files. In addition to the previously shown screens, there are files for certain popup messages. The layout files are the same for everything (for example, all the lessons have the same Lesson Menus) but their texts change according to the Java code which will be explained later.
  It is also important to note that to simplify the design process a style was used and implemented to all components. This style adheres to a certain color scheme with primary, secondary, and accent colors. This will help with later implementation of two themes: light and dark.
- As for the model, unfortunately some issues were faced with implementing a JSON Object for the lesson details. Therefore, these details are temporarily modeled in the form of a text file. There are three main text files: purpose and objective, notes, and questions.
  The purpose and objectives contains the purposes of the language and objectives of the course and it is used in the Lesson Menu. The first line has the title, the second the purpose, and the third the objective.
  The notes file has the course notes, which are used in the Lesson Course activity. Each two lines represent an entire page. The first is the title, and the second is the explanation.
  The questions file has the questions, the choices, and the correct answer for the Lesson Test activity. Each seven lines represent a block. The first is the lessonId, the second is the question, then 4 lines for the choices, and the last one is the correct answer.
- Now, how are the model and view connected? The key is the lessonID. As mentioned each lesson has its own unique ID. This ID is used to mark and separate the blocks. In the Java code, the pressed button has an ID which is carried to the next activity. At the first, a while loop works through the text file and compares the ID to the lines. If the ID is equal, the required lines are taken and stored in an array to be later used, else they are skipped and the loop repeats.
  For example, for the purposes and objectives file, the loop would take the first line and compare it with the given ID. If it is equal, then the first line is taken as the title, the second as purpose and third is the objective. If not, then the three lines are skipped and the following line representing the ID is taken and compared. Since all blocks are modeled like this, there should be no problems in taking the wrong lines.
  It is also important to note that Java would just take the text as is. Therefore, special characters are inserted in the text (like \n and \t) which are then replaced with the proper syntax to get the result.

For the PHP environment, the implementation was done on WampServer 3.2.6 as database and Visual Studio Code 1.85.0 for the coding.

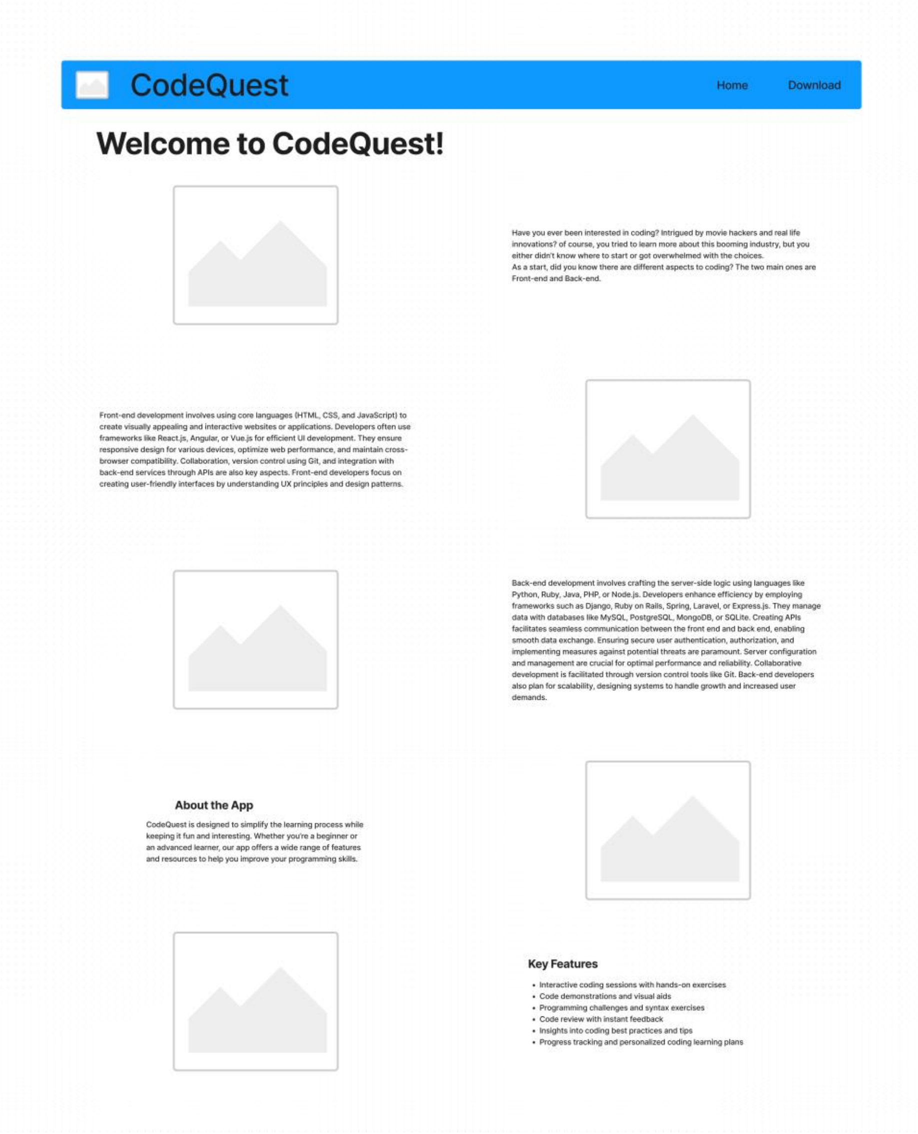
- Here we return to the idea of an online and offline aspect. Offline, there is a UserManager class which would store the user profile information which would be used later throughout

the application. The purpose of the PHP code is to initially get that information and the Java code would take it and store it. Throughout the application any changes applied to the database through the PHP is also applied to the offline UserManager.
The PHP code would also get information related to the courses being taken as in lessonId and title and relay it to the Java code. In turn, the Java would generate a certain amount of buttons based on this information.

- Some issues were faced regarding the echoing of the results, where Java would not properly interpret the result. Most of these were resolved by changing the variable type echoed, while some are still being checked.

3.2. React Website Implementation

The React Website was implmented using Node.js v21.5.0 and Visual Studio Code 1.85.0.



Welcome to CodeQuest!

Have you ever been interested in coding? Intrigued by movie hackers and real life innovations? of course, you tried to learn more about this booming industry, but you either didn't know where to start or got overwhelmed with the choices.
As a start, did you know there are different aspects to coding? The two main ones are Front-end and Back-end.

Front-end development involves using core languages (HTML, CSS, and JavaScript) to create visually appealing and interactive websites or applications. Developers often use frameworks like React.js, Angular, or Vue.js for efficient UI development. They ensure responsive design for various devices, optimize web performance, and maintain cross-browser compatibility. Collaboration, version control using Git, and integration with back-end services through APIs are also key aspects. Front-end developers focus on creating user-friendly interfaces by understanding UX principles and design patterns.

Back-end development involves crafting the server-side logic using languages like Python, Ruby, Java, PHP, or Node.js. Developers enhance efficiency by employing frameworks such as Django, Ruby on Rails, Spring, Laravel, or Express.js. They manage data with databases like MySQL, PostgreSQL, MongoDB, or SQLite. Creating APIs facilitates seamless communication between the front end and back end, enabling smooth data exchange. Ensuring secure user authentication, authorization, and implementing measures against potential threats are paramount. Server configuration and management are crucial for optimal performance and reliability. Collaborative development is facilitated through version control tools like Git. Back-end developers also plan for scalability, designing systems to handle growth and increased user demands.

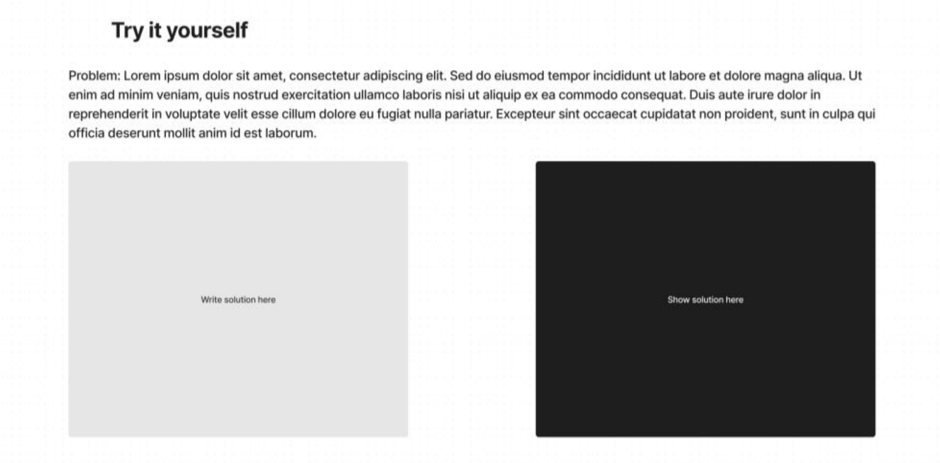
About the App

CodeQuest is designed to simplify the learning process while keeping it fun and interesting. Whether you're a beginner or an advanced learner, our app offers a wide range of features and resources to help you improve your programming skills.

Key Features

- Interactive coding sessions with hands-on exercises
- Code demonstrations and visual aids
- Programming challenges and syntax exercises
- Code review with instant feedback
- Insights into coding best practices and tips
- Progress tracking and personalized coding learning plans

The main purpose behind the website to be a complement to the application, a front to display the key points of the work being done. It features information regarding the different aspects of programming. First, it focuses on Front-end and Back-end development, and then it delves deeper into the languages behind each aspect.

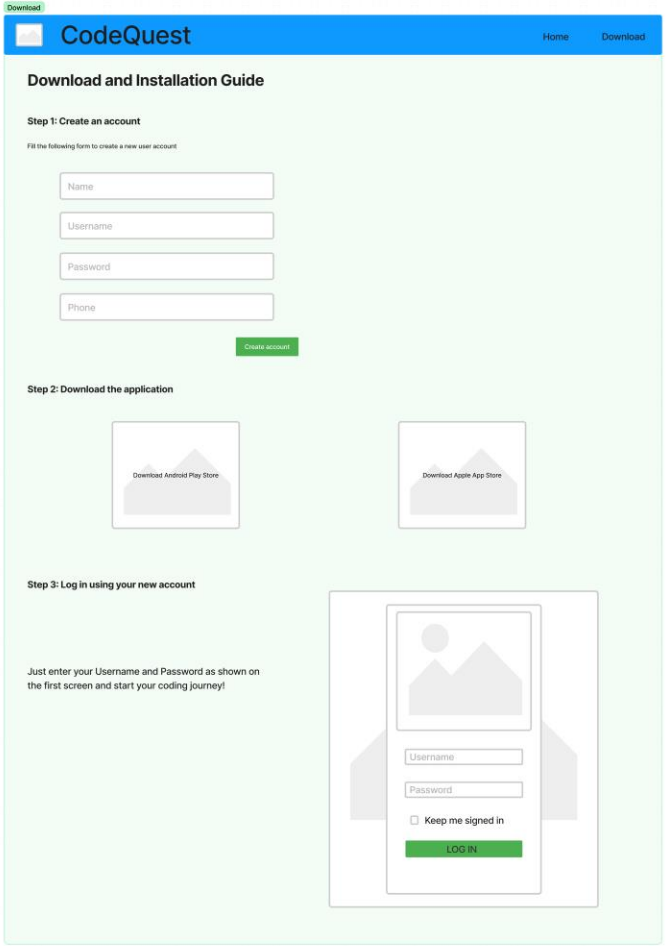


Try it yourself

Problem: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

In the same page at the end, we see some problems to be solved. Beginners who know the basics of programming could use this. This would even show the solution and the possible applications and developments.

Of course, there would be a page dedicated to the application. This would contain a Create Account prompt to get started. Then, there would be the link to download it on the mobile device.



Download and Installation Guide

Step 1: Create an account

Fill the following form to create a new user account

Step 2: Download the application

Step 3: Log in using your new account

Just enter your Username and Password as shown on the first screen and start your coding journey!

4. Conclusion

Overall, my application could be considered important in simplifying the journey one goes through in programming. It would prevent excited people from becoming overwhelmed and suddenly lose interest in something they could have high potential in.

For me, it is important that the UI/UX is simple enough and the information strong to keep users attached to and interested in learning. There should be additions to the courses and what they teach in terms of method. Currently, a user only learns through theoretical reading and checks progress through a quiz. There would also be practical practice where users would read a problem, test a solution, and see the results in real time.

In terms of application, it is important that the code remains clear through syntax and comments to help future developers in understanding the importance of each point. There will also be improvements to storage, where the objective, notes, and questions banks could be stored on a cloud service to facilitate information editing. In addition, as the application grows, there will be enhancement of speed in data retrieval from the online database.