



Spring 2022 ITS Quiz App

Neha Lalani, Chinmayi Kompella, Sakshi Deshpande,
Abdulaziz Memesh, Khushi Magiawala, Krishan Patel



Project Overview

Problem/Motivation:

Existing ITS tools are not targeted for mobile devices or on-the-go studying.

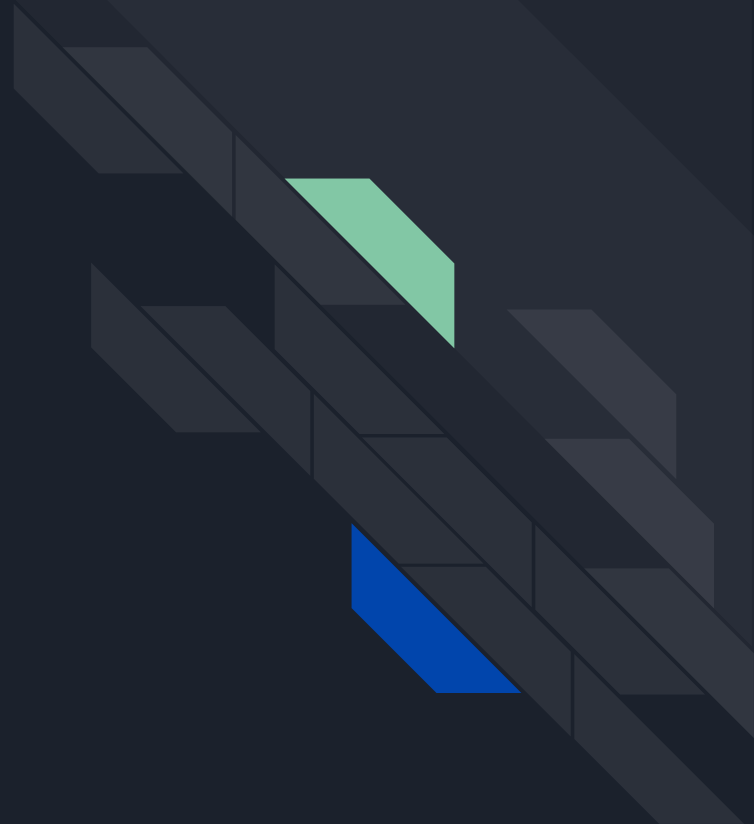


Project Overview

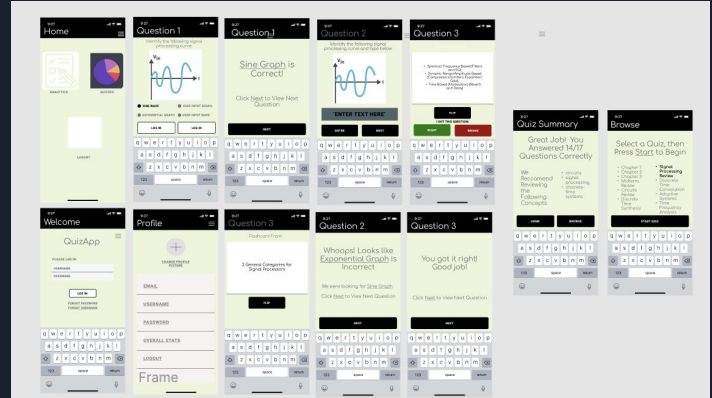
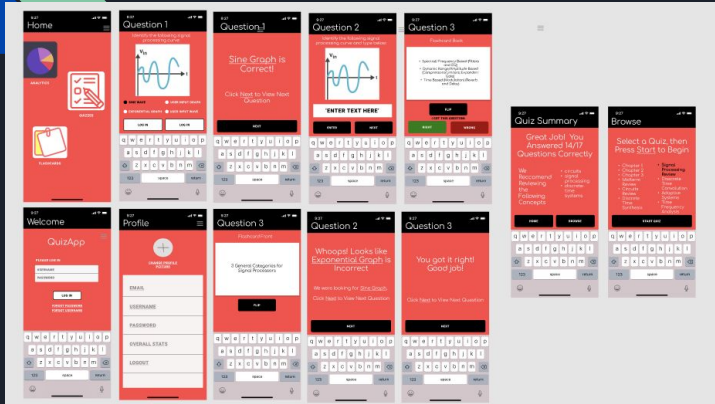
Semester Goals:

- Continue the development of the mobile quiz app
 - Gain mobile-app development experience (especially for new members)
 - Refactor UI for a more attractive front-end
 - **New Features:**
 - i. Create screens and logic to handle credential validation and multiple users
 - ii. Answer Validation
 - iii. Hint Screen
 - iv. Quiz Search Functionality
 - v. Toggle for reviewing questions
 - vi. Analytics Screen
- Lay groundwork for future integration with Knowledge Tracing
 - Develop **API** to handle QuizApp-MongoDB communication
 - Integrate QuizApp with API

Refactoring UI



Color Usage and Standardization



Help button for chat bot



chat bot potential idea



Help button for chat bot



chat bot potential idea



Frame 1



Why This UI?

- Personality of the brand
- Set the basic mood, tone, concept, and connotation for a brand or product
- Color theory
- Logo Color

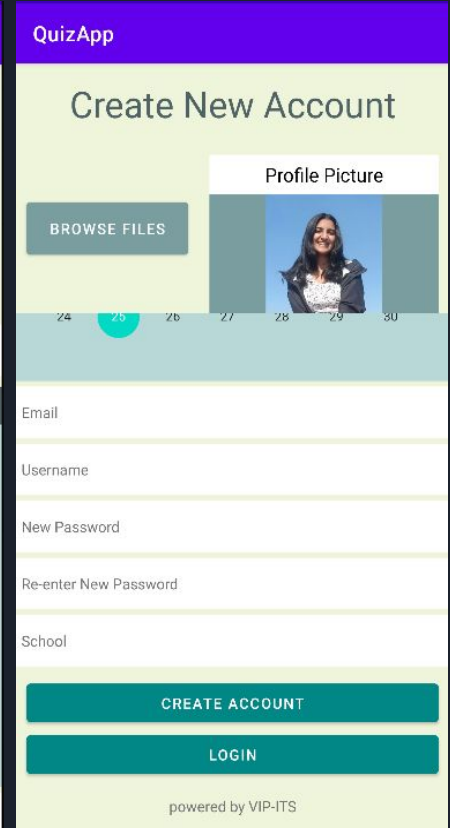
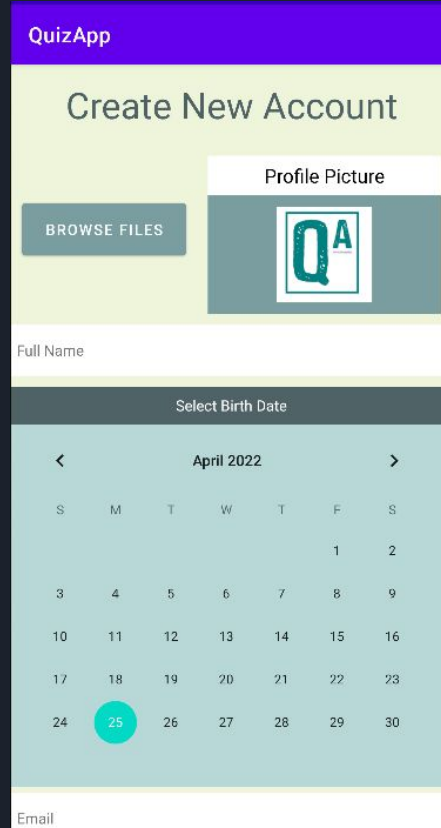
color [Ⓐ]	Meaning & key words [Ⓐ]	Usage pattern [Ⓐ]
Black [Ⓐ]	Luxury, value, elegance, sophistication, power, classic [Ⓐ]	E-commerce, high-value sites [Ⓐ]
red [Ⓐ]	youthful , power, stunning, stimulating, energizing, passion [Ⓐ]	Warnings, food, notices [Ⓐ]
yellow [Ⓐ]	Warning, happiness, fun, Playfulness, enthusiasm, wisdom [Ⓐ]	Warning signs, traffic signals [Ⓐ]
green [Ⓐ]	nature, environmental, stability, growth, balance, stabilization [Ⓐ]	Environment, outdoor products. [Ⓐ]
gray [Ⓐ]	Traditional, professional, formality, neutrality, melancholy [Ⓐ]	Most used in website [Ⓐ]
orange [Ⓐ]	Haste, impulse, fun, activity, competition, confidence, movement [Ⓐ]	Sports, children, cartoon, logo [Ⓐ]
white [Ⓐ]	freedom, spaciousness, Clean, simplicity, virtue, purity, innocence [Ⓐ]	background color, web page [Ⓐ]
purple [Ⓐ]	Luxurious, romance, mystery, royalty, lavishness, wealth [Ⓐ]	Fashion, luxurious commodities [Ⓐ]
blue [Ⓐ]	trust, peace, order, loyalty, calmness, openness, safety, calmness [Ⓐ]	Common used, social website [Ⓐ]
ivory [Ⓐ]	elegance, simplicity, comfort, Clean, reliability, safety [Ⓐ]	Coffee Website, warm product [Ⓐ]

Login Functionality



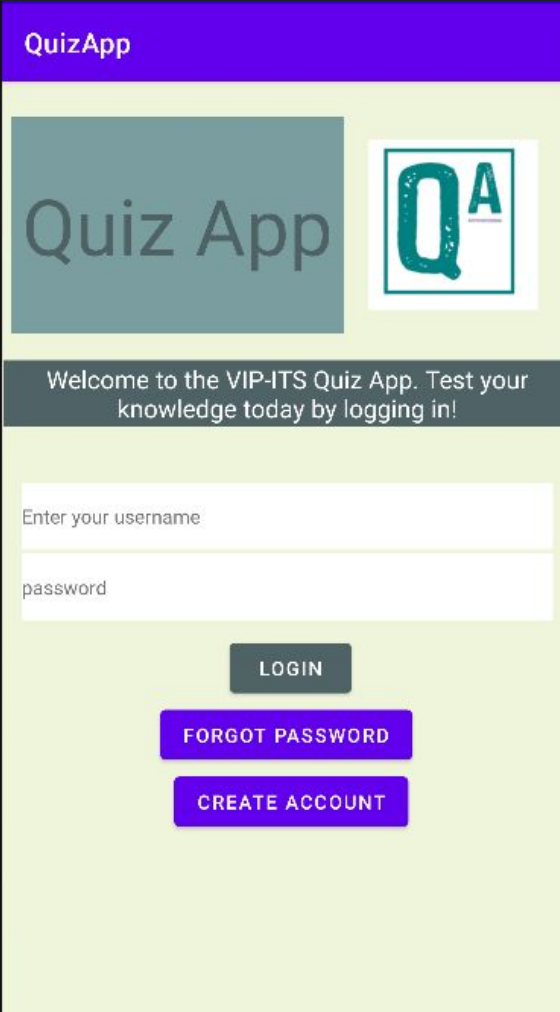
Create Account Screen

- First screen in application flow
- Allows user to input personal information and create account
- Account Validation
 - Cannot create the same account twice
 - Checked using username attribute in the back-end
 - Has to fill out required fields
 - Full Name
 - Username
 - Password



Login Screen

- Can access this screen directly from 'Create Account' screen
- Allows you to use credentials and load up user information for a more personalized experience
- Credential Validation
 - Checks if the username and password pair exists and loads that user up in the User object



QuizApp

Quiz App

QA

Welcome to the VIP-ITS Quiz App. Test your knowledge today by logging in!

Enter your username

password

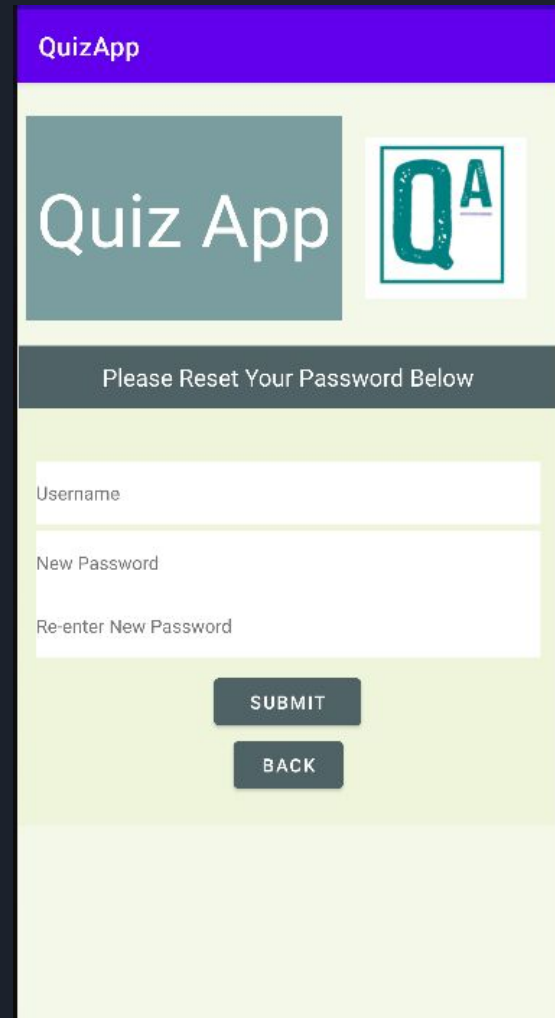
LOGIN

FORGOT PASSWORD

CREATE ACCOUNT

Forgot Credentials Screen

- Can access this screen directly from 'Login' screen
- Gives the user a chance to reset their password in case they forget
- Back-end uses username to find the account and change credentials



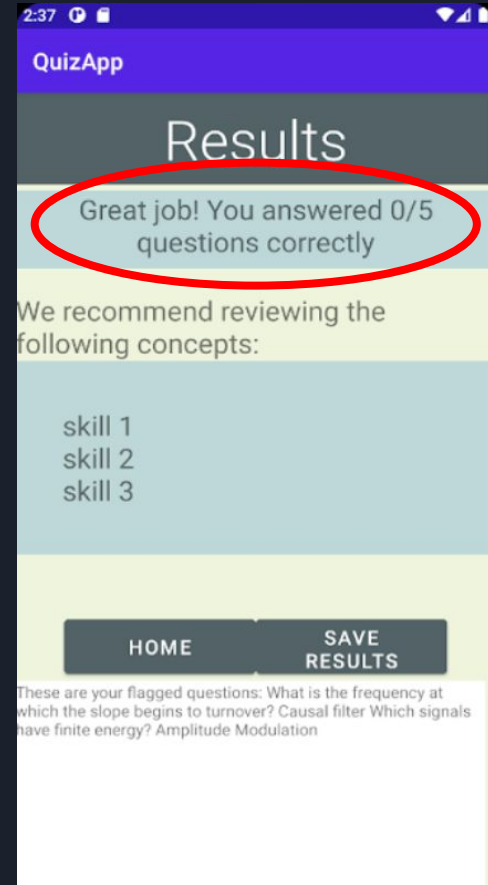
The screenshot shows the 'Forgot Credentials' screen of the Quiz App. At the top, there is a purple header with the text 'QuizApp'. Below the header, the screen is divided into two main sections. The left section features the text 'Quiz App' in a large, white, sans-serif font on a dark teal background. To the right of this text is a square logo with a white border containing the letters 'QA' in a stylized, teal font. Below these elements is a dark teal banner with the white text 'Please Reset Your Password Below'. Underneath the banner, there are three white input fields stacked vertically, each with a dark teal label: 'Username', 'New Password', and 'Re-enter New Password'. At the bottom of the form area, there are two dark teal buttons with white text: 'SUBMIT' and 'BACK'.

New Feature Development



Answer Validation

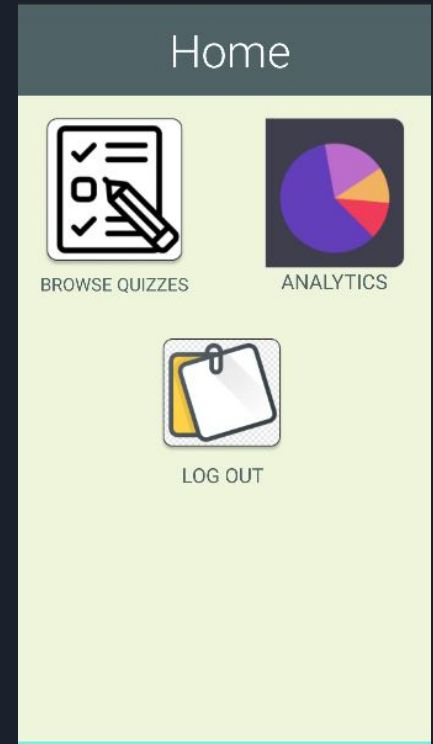
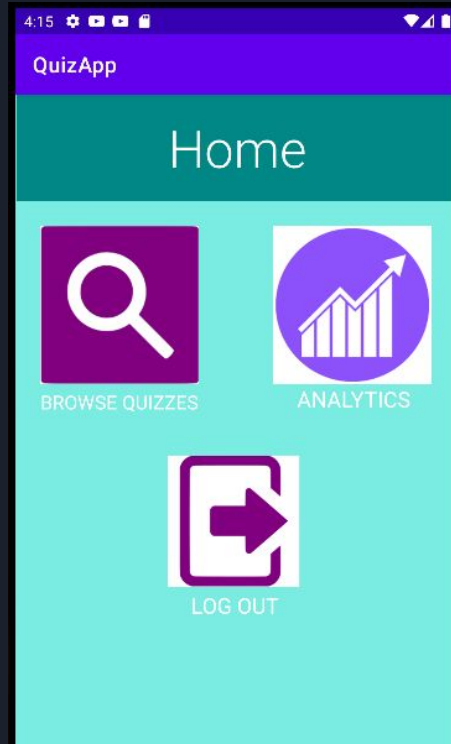
- Grade as you go on within the quiz so results screen shows quiz score
- Answer before moving on
 - Future implementation could be you can flag and go back to a question rather than forcing an answer





Changing
text buttons
into image
buttons

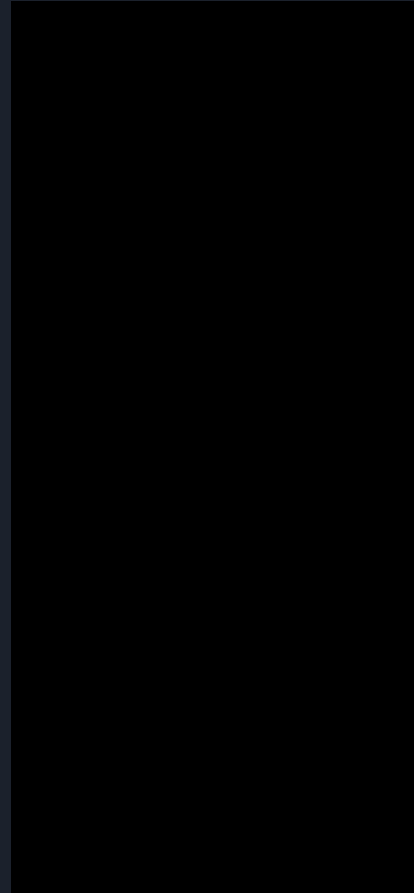
Now





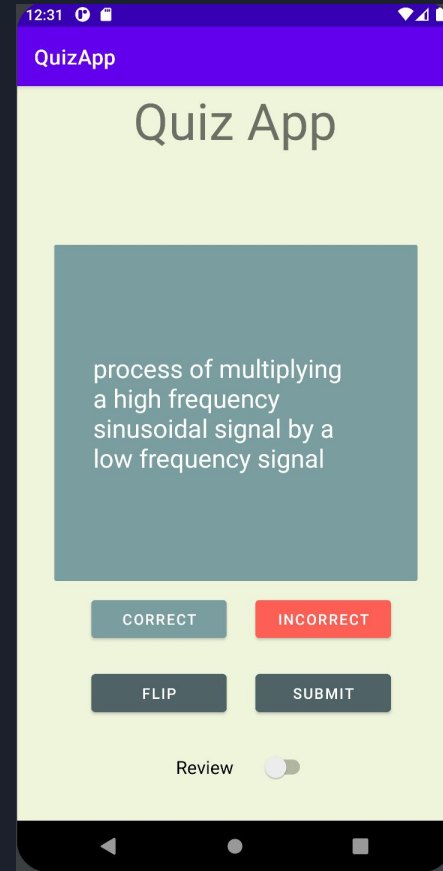
Search Bar

- Usefulness
- UI Components
- Future Development



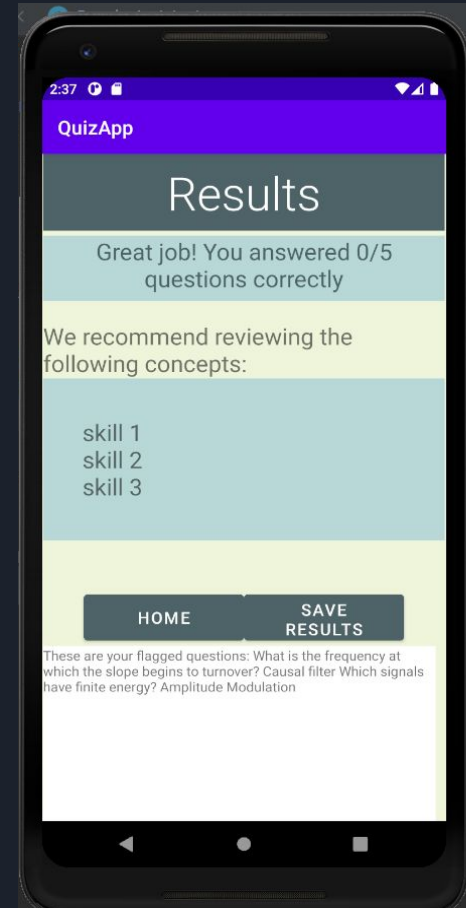
Toggle Button

- Saves the question that users want to review



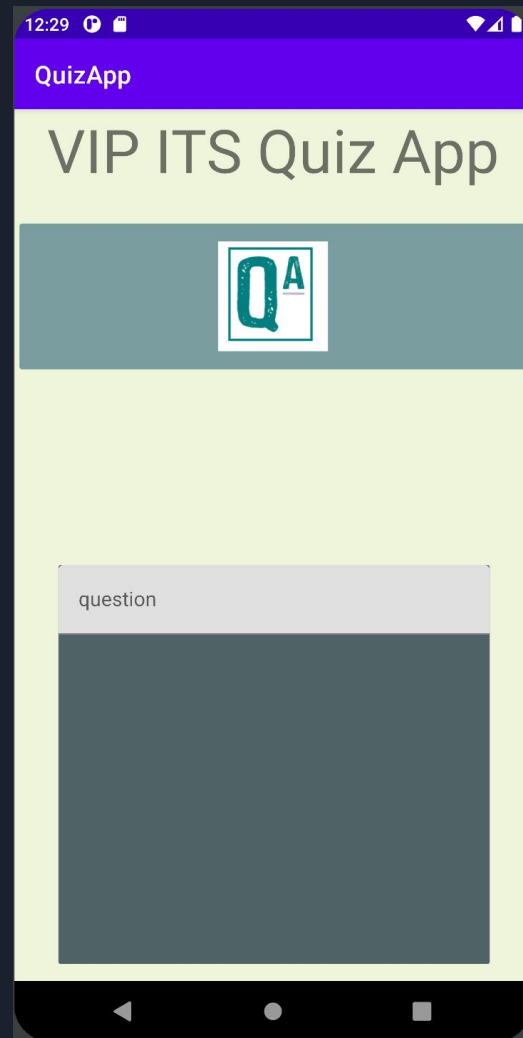
Toggle Button

- Final review screen with a list of the questions that users toggled



Help/Hint Page

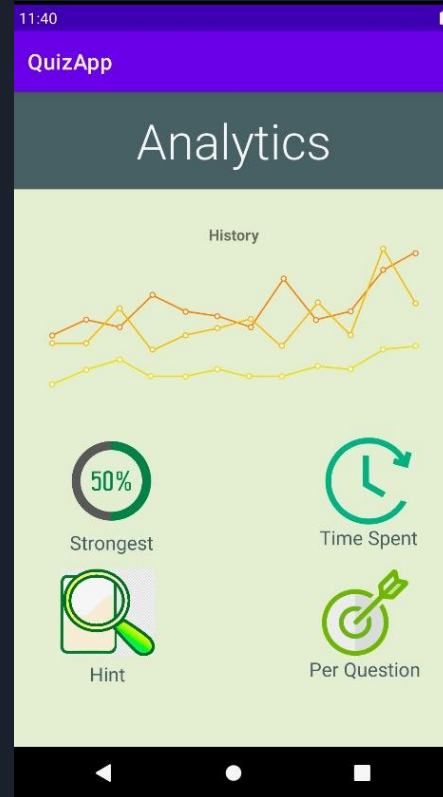
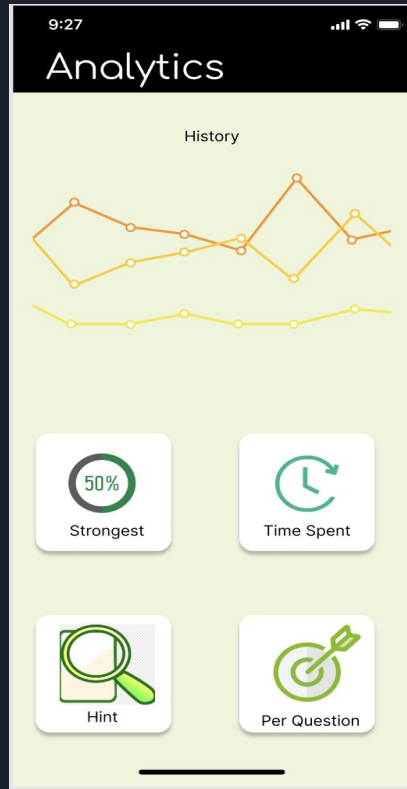
- Users can access this page when they need hints or help on a question



Analytics + Knowledge Tracing



Analytics





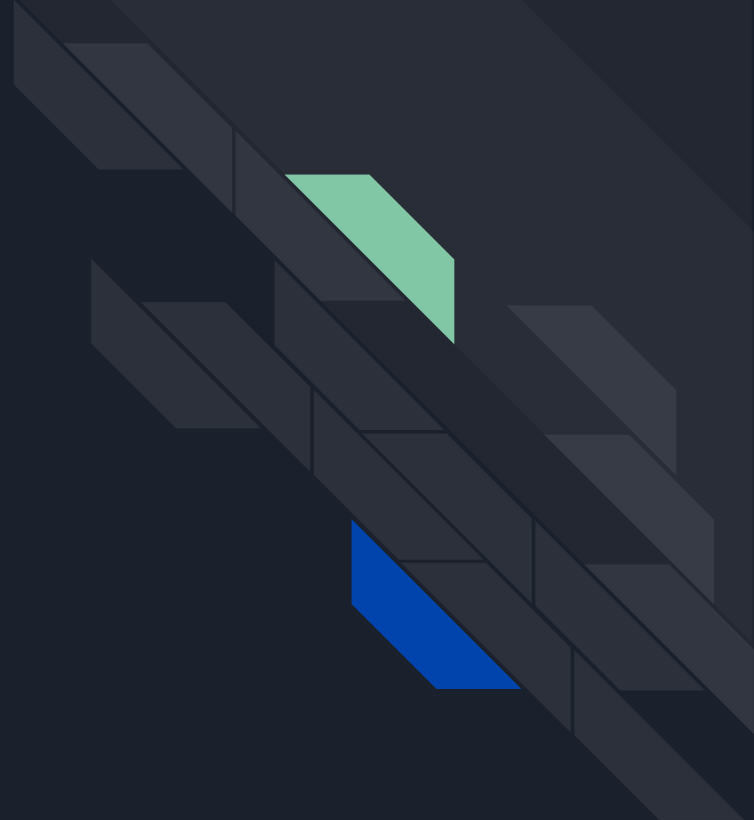
Future Front-end Goals

Minor Developments

1. Display analytics from ML model on Analytics Screen
2. Minor UI refactoring changes (still some color inconsistency)

For the most part, the UI is adequate enough for users to take quizzes and get results. Now, most of the work lies in creating API endpoints and KT ML logistics.

Backend





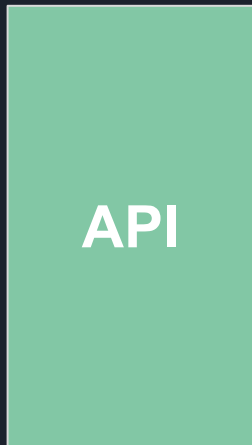
Goals

- Create an API to interface with MongoDB and Knowledge Tracing model
- Develop QuizApp to API integration
- Standardize QuizApp data to match data stored in MongoDB
- Process and convert data to/from JSON

Flow



Request/Store Data



Request
Recommendation/Store
Student Data



Retrieve query data



Database



mongoDB

- Store app data such as user credentials, questions, and user's quiz performance
- MongoDB Atlas Cluster (cloud-based)
 - have user details tied to their account, rather than the device itself
 - allow admin access to questions currently stored in the database
 - Simpler to query and process data than with SQL

Data Cleanup

- Hints attribute changed
 - Rewording of some attributes so they aren't placeholders anymore
- Extra attributes of the short answer, flashcard, and multiple choice questions stored in question

```
_id: ObjectId('621c4ccb07a52bae478ff402')
quizId: 0
questionText: "The period T of a periodic signal x(t) is"
correctAnswer: "The values of the signal repeat every T seconds"
topic: "topic1"
questionType: 1
backText: ""
> incorrectChoices: Array
  __v: 0
  questionId: "q1"
  hint: "Another way to describe a period is any one full pattern in a cycle."
```

```
_id: ObjectId('621c4dd707a52bae478ff404')
quizId: 0
questionText: "What is a spectrum?"
correctAnswer: "a graphical presentation of the complex amplitude for each frequency c..."
topic: "topic2"
questionType: 1
backText: ""
> incorrectChoices: Array
  __v: 0
  questionId: "q2"
  hint: "A spectrum has something to do with the amplitude of a signal. "
```

API Tech Stack



Runtime
Environment



Handles efficient
request routing,
response formulation



ODM tool for schema
enforcing, query
simplification



Testing tool for
individual API endpoints



API Endpoints

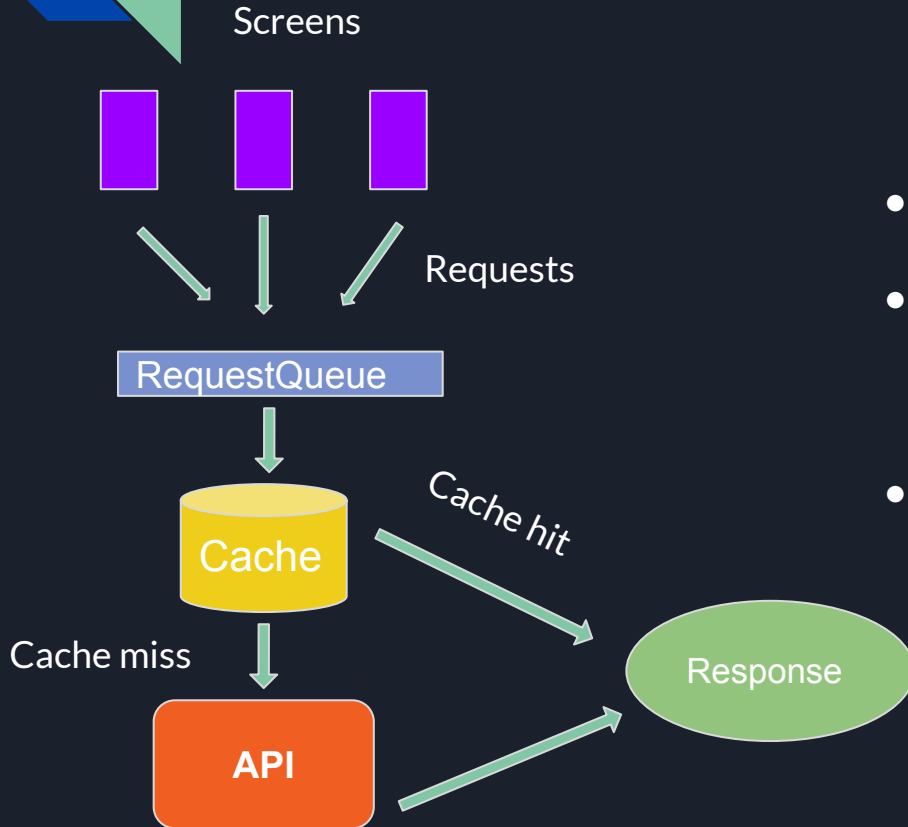
- Retrieve Quiz Questions
 - readRandom: sends a list of 5 questions of random question types and topics
 - readRecommended: sends a list of 5 questions of recommended topic
- Store Quiz Results
 - recordAnswers
- User Credentials
 - getUser
 - setUsername
 - setPassword
 - recordUser



Proposed API-KT Integration

- `Update_database`: will be called every time the app requests to save results
- `Predict`: called when `readRandom` endpoint is called to get a skill to recommend

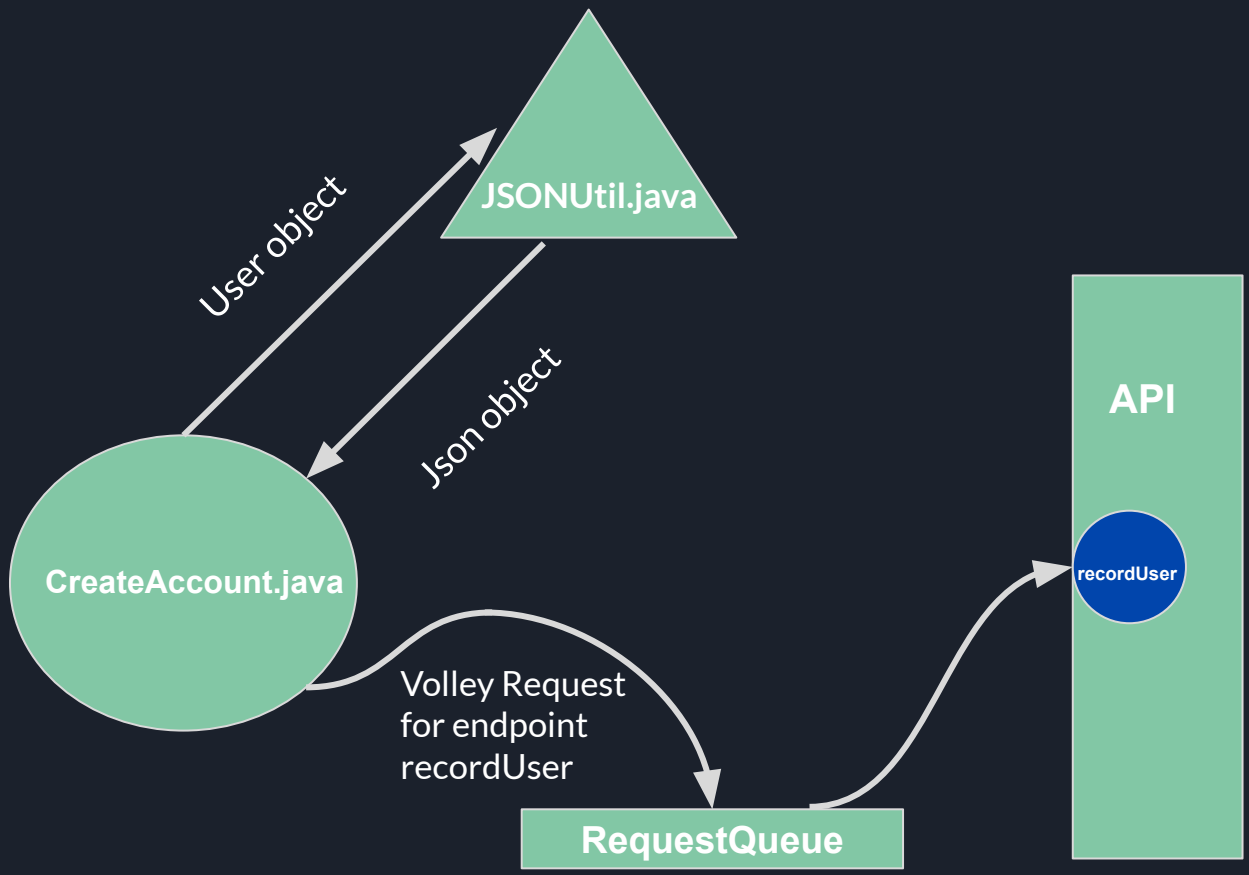
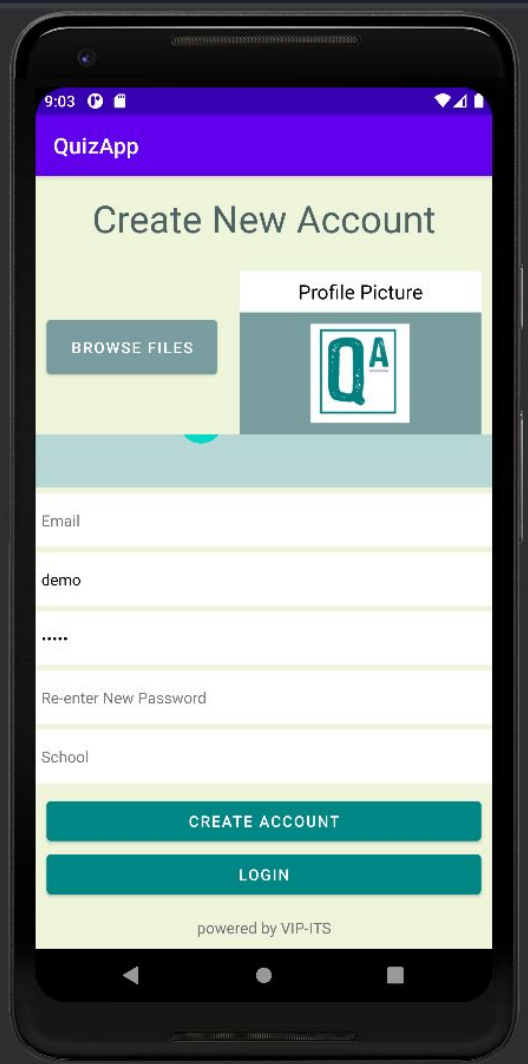
QuizApp-API Integration: Volley



- Used Volley, an HTTP library, to make requests from the app to specific API endpoints
- Ensured synchronous behaviour by handling responses in response handler ONLY
 - Can block main UI thread, but can solve this using a loading icon
- Singleton RequestQueue that queues each app request in FIFO order

Login Walkthrough





```
_id: ObjectId('62689e8e094103b27e183cf0')
userId: "demo"
password: "demo1"
name: "chin"
__v: 0
```

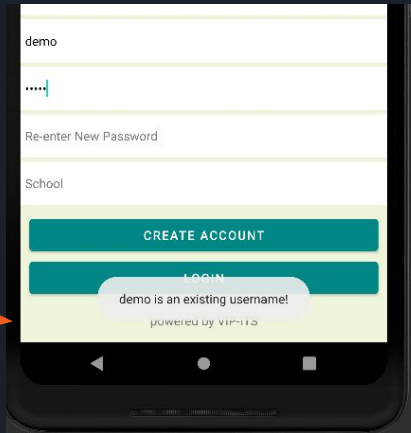
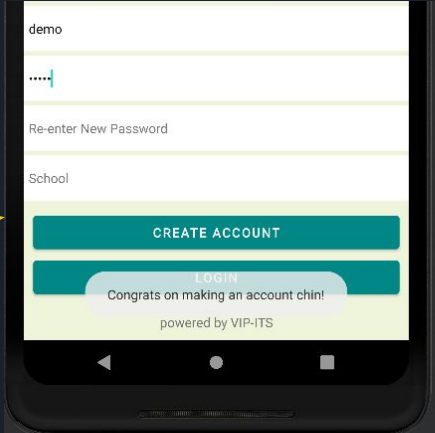
recordUser

Username
unique

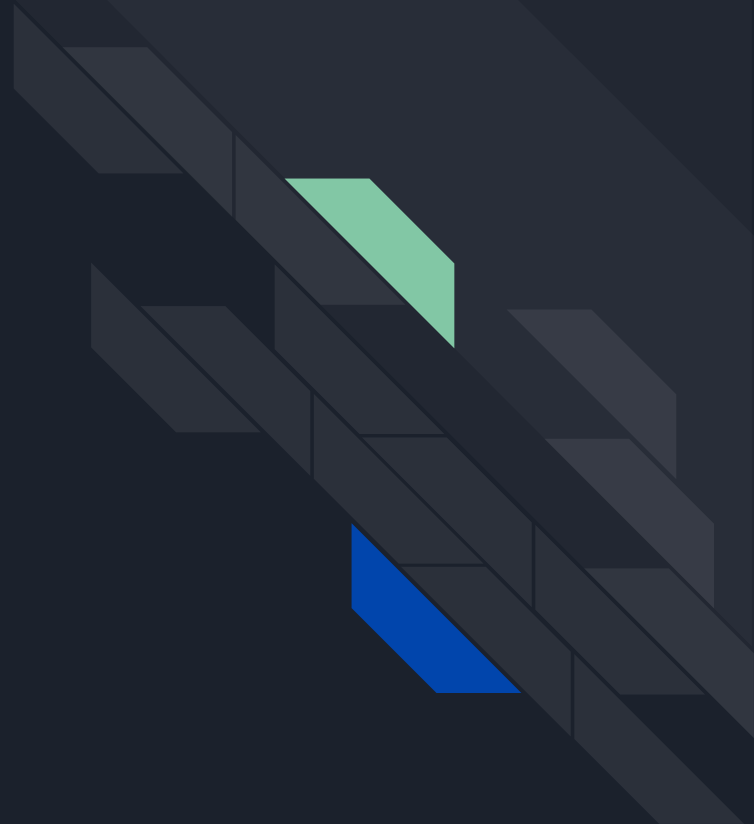
Send
status(200)

Send
status(400)

CreateAccount.java



Demo





Knowledge Tracing Team: Next Steps

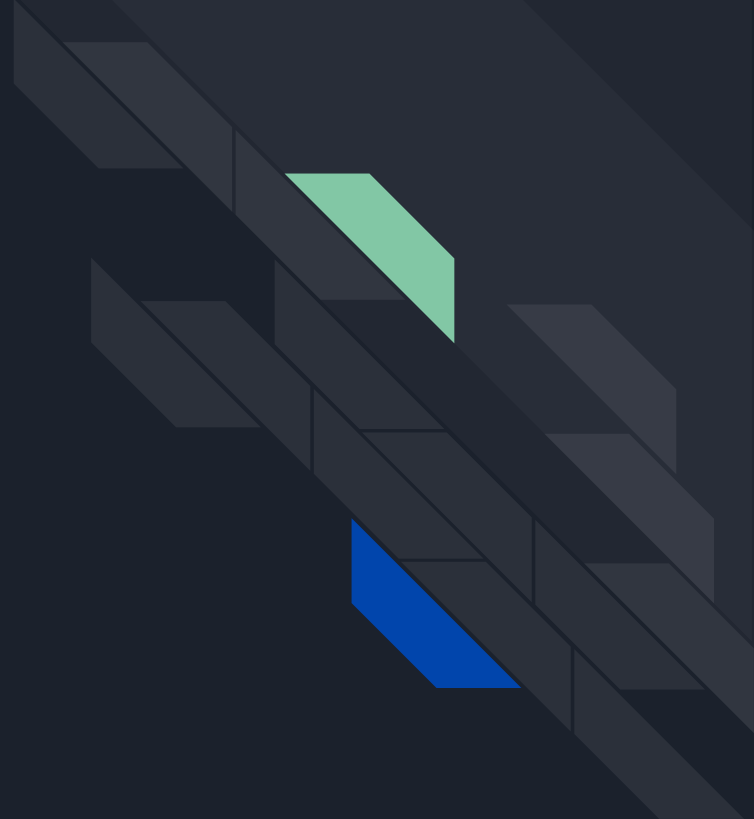
- Now that we have an API, we need to integrate KT's API with our API
- Need to train the KT ML model on the ECE Textbook question dataset
 - Requires question and answer collection for each question type
 - Dataset cleaning
- Run some prediction and results validation on the ML model to see how accurate it's quiz suggestions are



Some Future Features

- Integration with Knowledge Tracing API
- Analytics screen
 - Data aggregated during quiz session
 - Topic strength
 - Time spent per question
 - Hints used
- Ability for students to create their own quizzes
- Personalized hints based on past performance
- Set study times and send notifications to user

Thank you



Questions?

