

# LogiQ App

## Specification Document

## 1 Overview

LogiQ is a mobile quiz app that helps users practice formal logic in an interactive way. It is designed for logic learners and provides real-time feedback, detailed performance statistics, and support for both propositional and first-order logic. The app includes multiple types of logic tasks.

## 2 Formal Logic

### 2.1 Scope

This app supports both propositional logic and first-order logic.

### 2.2 Task Types

- **Truth Table:** Tasks involve evaluating the truth value of a logic expression under given truth assignments. *Example: If  $p = \text{true}$  and  $q = \text{false}$ , what is the value of  $p \wedge q$ ?*
- **Logical Equivalence:** Tasks ask whether two formulas are logically equivalent. *Example: Which of the following formulas is logically equivalent to  $\neg(p \vee q)$ ?*
- **Inference Validity:** Given one or more premises and a conclusion, the user must determine whether the inference is logically valid. *Example: Premises:  $\forall x(A(x) \rightarrow B(x))$ ,  $A(a)$ ; Conclusion:  $B(a)$ . Is the inference valid?*

### 2.3 Task Formats and Difficulty

The app supports **single-choice**, **multiple-choice**, and **true/false** formats. Each task is assigned one of three difficulty levels: **easy**, **medium**, or **hard**, based on factors such as the number of variables and the depth of logical nesting.

## 3 Frontend and App Features

### 3.1 Screens

- **Login and Register:** This screen allows users to log in or register using their email address.
- **Home:** The central hub of the app. It displays general information and provides buttons to start a random quiz or to select a specific task type and difficulty level.
- **Quiz:** Presents one task at a time, including the question statement and answer options. A timer and progress indicator are displayed. After completing a quiz round, the screen shows feedback.
- **History:** This screen records a history of quizzes completed by the user.
- **User:** This screen combines user information and statistics. Users can view their accuracy rate, average solving time, and error distribution. All data is stored persistently.

- **Guide:** A dedicated screen that explains the app's functionality with annotated screenshots and instructional text.

### 3.2 Features

- **Timer:** A timer is shown during each quiz task to record how long the user takes to complete the question. The completion time is stored and used in statistical analysis.
- **Light and Dark Mode:** The app supports light and dark themes. Users can switch manually.
- **Different Task Types and Difficulty Levels:** As mentioned in 2.2 and 2.3
- **User Management:** All user-related information and personalized statistics are synchronized. Passwords are securely stored.

## 4 Server-Side and Task Generation

### 4.1 Server for App

This part exposes RESTful APIs that support the core functionality of the app.

For instance, GET /tasks – returns a random batch of logic tasks.

### 4.2 Admin Interface

The server includes an HTML-based admin interface that supports **management of tasks**, including viewing, adding, and deleting tasks, as well as **viewing statistics** such as the success rate of each task.

### 4.3 Database

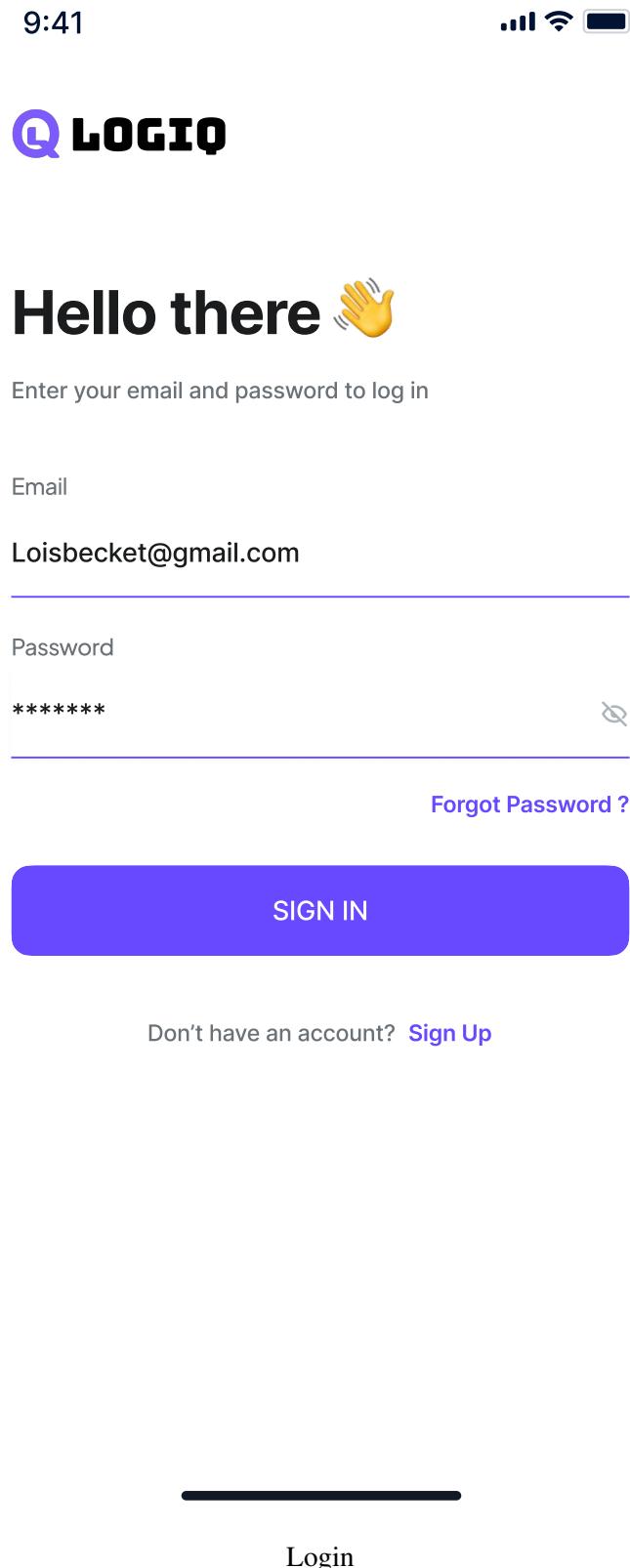
All data is stored in a local MySQL database running in a Docker container. The stored data includes logic tasks, personalized and anonymous statistics, authentication information and others.

### 4.4 Automatic Task Generation

The server includes a dedicated logic task generator written in Go. It combines two complementary strategies.

- **Template-Based Generation:** A collection of task templates is defined in advance, representing common logical patterns and reasoning structures. Each template can be instantiated by substituting variables, predicates, or connectives. This ensures that the tasks generated are controlled and of high quality.
- **Dynamic Generation:** In order to generate a large number of unique tasks, the generator can also dynamically generate tasks by constructing syntax trees of logical expressions and randomly combining operators and variables.

## Appendix: User Interface Sketches



9:41



# Sign up

Create an account to continue

Full Name

Ben

Email

ben@gmail.com

Set Password

\*\*\*\*\*



Continue

Already have an account? [Login](#)

Register

9:41



Hi, Ben 🙌



time to unlock some logic!

**Quiz**

Random Tasks

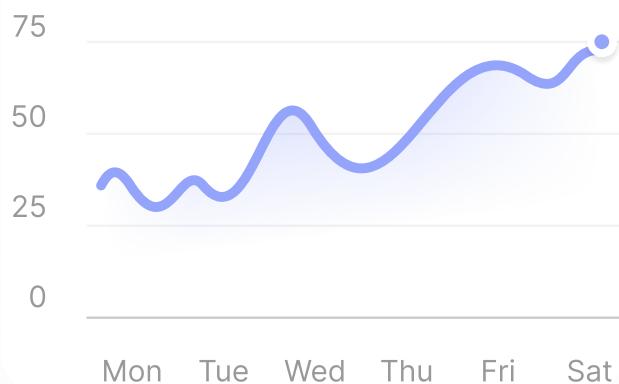


Topic Practice



By Diffic

&gt;

**Performance**[View All](#)

Home



History



Guide



User

Home

9:41



00:14

8/10

## Question

Which of the following formulas is logically equivalent to  $\neg(p \vee q)$ ?

A.  $p \vee q$ B.  $p \vee q$ C.  $p \vee q$ D.  $p \vee q$ 

&lt;

Next

Quiz

9:41



## Q LOGIQ



Random Tasks

8/10 Correct · 4m12s



Topic Practice

8/10 Correct · 4m12s



By Difficulty

8/10 Correct · 4m12s



Custom Quiz

8/10 Correct · 4m12s



Home



History



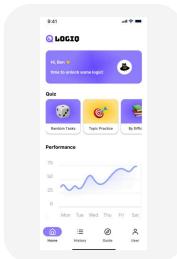
Guide



User

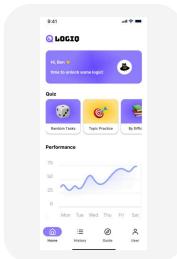
History

9:41



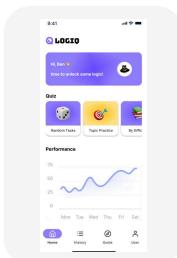
### Function or Screen

Instructional text text text text  
text text text text text text text  
text text text text.....



### Function or Screen

Instructional text text text text  
text text text text text text text  
text text text text.....



### Function or Screen

Instructional text text text text  
text text text text text text text  
text text text text.....



Home



History



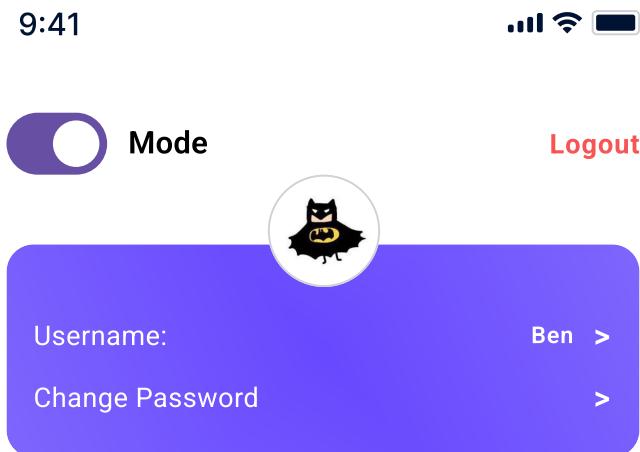
Guide



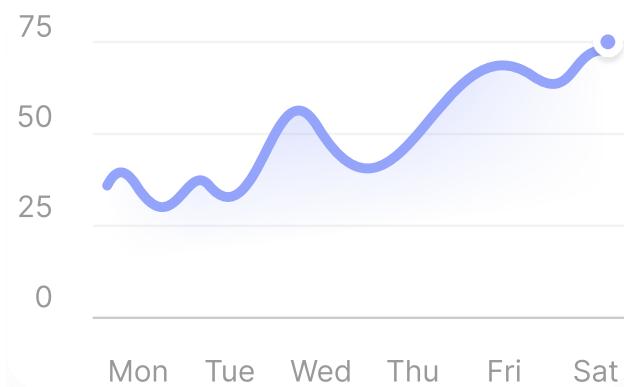
User

---

Guide



### Accuracy Rate



### Error Distribution

