

플러터와 장고를 이용한 한국사 퀴즈 앱/서버 개발



과목명 컴퓨터공학종합설계1

교수님 오승현교수님

학번 이름 김단 2019213997

오아연 2020213297

제출일 2023/05/12

목 차

1. 서론
 2. 개발 환경 및 개발 내용
 3. 실험 및 결과
 4. 결론
- 부록: 소스 코드

1. 서론

바쁜 직장인이나 학생들이 자투리 시간에도 간편하게 앱을 이용하여 한국사능력검정시험 자격증을 준비하는 데 도움을 줄 수 있는 퀴즈 앱을 개발하고자 했다. 자격증의 경우 일정 점수만 넘기면 합격인 형식이라 언제 어디서든 손쉽게 퀴즈를 통해 시험을 치고 자신이 합격할 수 있는 점수인지 확인할 수 있도록 하는 앱을 개발하면 많은 수험생들이 편리하고 유용하게 사용할 수 있을 것이라 생각했다.

한국사능력검정시험은 한 나라의 국민으로서 가져야 하는 기본적 역사적 소양을 측정하고, 역사에 대한 전 국민적 공감대를 형성하기 위한 시험으로 한국사 학습능력을 측정할 수 있는 대표적인 시험에 속한다. 수많은 자격증 중 한국사능력검정시험 자격증으로 정한 이유는 공무원, 군무원, 공기업, 공공기관 등 다양한 정부처에 취업 시 가산점 혹은 응시자격이 부여하고 있기 때문에 많은 사람들이 필수로 공부를 해야 하는 자격시험에 속하기 때문이다. 또한 학습 기간이 대략 1-2개월 정도 소요되며 합격이 크게 어렵지 않아 모바일 퀴즈 앱에 적합하다고 생각하여 이 자격증으로 선택했다.

퀴즈앱의 개발범위는 홈 화면 (UI), 문제 리스트 (UI), 문제 풀기 화면 (UI), 문제 결과 화면 (UI) Django Rest Framework API 개발, 플러터에 API 연동을 하는 것까지 구현하려 한다. 이는 퀴즈 앱에 필수적인 기능이라 반드시 포함해야 한다. 로그인 기능과 타이머 기능 등 추가적인 기능도 계획하였으나 시간상 제외했다.

2. 개발 환경 및 개발 내용

2.1 개발 환경

퀴즈앱 개발에는 다양한 개발 환경이 사용되었는데 먼저 개발도구로는 Visual Studio Code(VSC)를 사용하였다. VSC는 점유율이 높고 다른 언어용 확장 프로그램을 많이 보유하고 있다. 또한 깃 지원과 같은 내장 소스 코드 관리 기능을 갖추고 있어 VSC로 선택했다.

Flutter는 구글이 개발한 앱/웹/데스크톱 앱 개발 프레임워크이다. 초창기에는 Dart라는 생소한 언어로 인한 어려움이 있었지만, React Native를 가뿐히 뛰어넘는 앱 성능과 개발 속도를 자랑하며 큰 인기를 끌고 있다. 하나의 코드로 iOS, Android, macOS, Window, web 모두 동작 가능하며 상업적인 측면에서 많은 대기업들이 Flutter를 기대하고 있어 Flutter로 앱을 개발하였다.

Flutter는 Dart 언어를 사용하는데 Flutter 팀에서 초기 언어를 선택할 때 12개 이상의 언어 중 UI 구축 방식이 Dart와 가장 일치하여 Dart를 선택하게 되었다고 한다. Dart 언어는 두 가지 컴파일 방법을 지원 (JIT + AOT) 하며 선언형 프로그래밍 방식을 사용하여 가독성과 유지보수 측면에서 좋다. 또한 핫 리로드 기능 등 여러 장점이 있다.

Django Rest Framework로 API 서버를 간단하게 만들었는데 Django Rest Framework는 Django 기반의 REST API 프레임워크이다. 다른 언어들에 비해 비교적 쉬운 Python과 장고를 기반으로 만든 프레임워크라 아주 간단하게 백엔드 API를 개발할 수 있다.

앱 배포를 위해 Heroku를 선택하였는데 Heroku로 서버를 돌리면 서버를 세팅하는 시간이 상당 부분 줄어든다. 서버 프로그램을 따로 설치할 필요도 없으며 소켓 프로그램도 필요 없다. 이 프로젝트는 Django Framework를 사용하기 때문에 Heroku Git Repository에 push 하면 끝이어서 간편하다. 다음 그림1은 시스템 구성도이다.

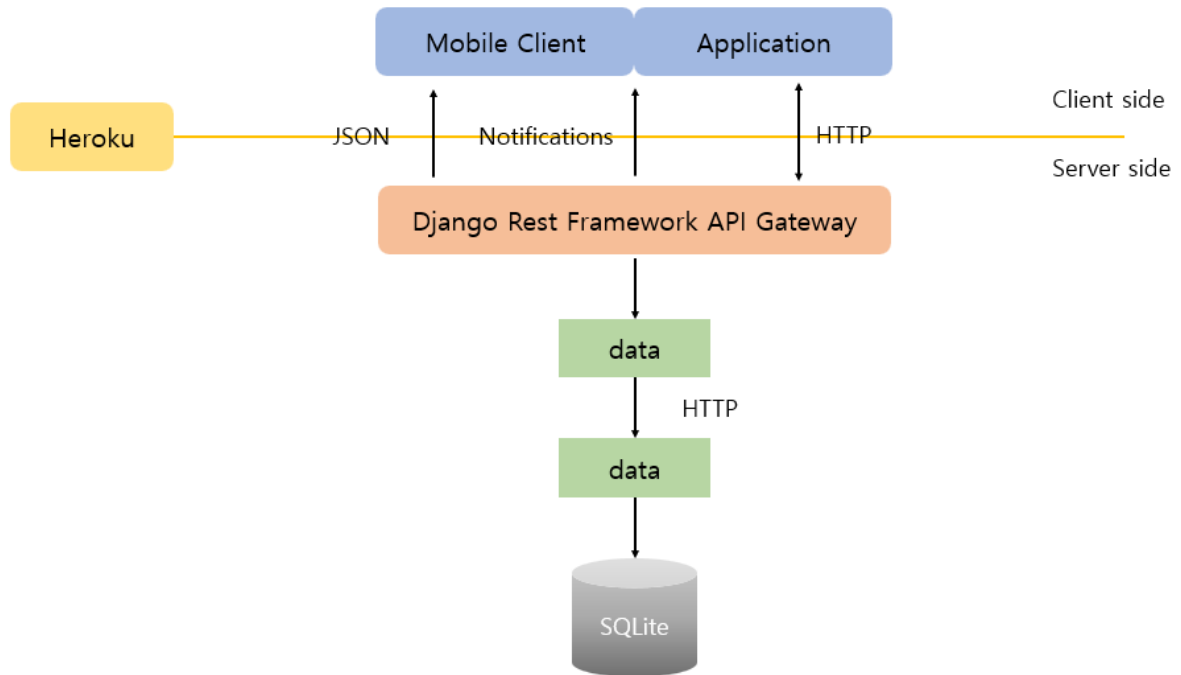


그림 1 - 퀴즈앱에 Django Rest Framework 기반의 API 서버를 연동해 퀴즈 데이터를 생성 후 Client side로 랜덤하게 출력할 수 있도록 하였고 이 프로젝트를 Heroku에 배포하여 관리자 계정에서 퀴즈를 관리할 수 있게 하였다.

2.2 개발 내용

1) 시스템의 기능 목록

- 처음 홈 화면으로 한국사능력검정시험의 기본, 심화로 나누어 사용자가 고를 수 있게 한다.
- 기본, 심화 각각 선택지에서 사용자가 원하는 회차를 선택할 수 있고 문제 풀이 화면으로 넘어간다.
- 기본적으로 퀴즈 데이터는 문제 풀기 화면에 랜덤하게 출력될 수 있도록 하였고 문제 풀기 화면은 기출문제 질문과 4개의 답변 선택지들로 구성하였다.
- 문제 풀이 화면에서 사용자는 보기를 고르고 다음 문제로 넘어간다. 이때, 사용자가 문제를 풀 시 다른 선택지를 누를 수 있게 하고 답을 하나 선택하기 전에는 다음 문제로 넘어가지 못하게 하였다. 답을 선택하면 보라색으로 어떤 답을 선택하였는지 확인하기 쉽게 구현하였다.
- 사용자가 문제를 다 풀고 나면 결과 화면으로 넘어간다. 이때 문제 결과 화면은 자신이 합격할 수 있는 점수인지 바로 확인할 수 있도록 '맞은 문제 / 전체 문제'로 결과 화면에 나타내었다. 그 다음, 홈으로 돌아가기 버튼을 누르면 기본, 심화를 선택할 수 있는 처음 화면으로 되돌아가서 사용자가 또다른 회차의 문제들을 풀 수 있도록 유도한다.
- 다시 돌아온 기출문제 선택지 화면에서 아이폰의 스와이프 기능이나 안드로이드의 백 버튼을 통해 이전에 풀었던 문제의 결과화면으로 되돌아가지 않도록 한다.

2) 구현한 시스템 기능

- Flutter의 StatefulWidget으로 너비, 높이부터 시작해서 프로젝트의 구조를 설정하였고 홈 화면을 만들었다.
- ListView를 기본, 심화 각각 생성해 사용자가 원하는 난이도와 회차를 선택할 수 있고 페이지를 ListView에서 문제 풀기 화면으로 페이지를 이동하게 Navigator 스택에 QuizScreen(문제 풀기 화면)을 push 해 준다. 즉, 버튼 클릭 시 문제들을 가져오는 방식을 사용했다.
- 문제들은 Python의 가상환경과 migration을 이용해 Django Rest Framework API

를 개발해 서버를 실행하고 관리자 계정 페이지를 만들어서 문제를 추가하는 등 Quiz 모델을 관리한다. 그리고 Serializer화(Django에 있는 문제 데이터들을 JSON 타입으로 바꿔서 API 통신이 되게끔 하는 것) 한 문제들을 불러와 Django 페이지에 출력되게 한다.

- 로컬로 실행돼 있으면 플러터 앱에서 접근하기 어렵기 때문에 실제 호스팅 서비스인 Heroku에 Django Rest Framework API를 배포한다. 마찬가지로 Heroku 사이트에서도 Django의 Quiz 모델을 관리할 수 있게 된다.
- 플러터에 API를 연동하여 API의 문제 데이터가 QuizScreen(문제 풀기 화면)에 랜덤함수를 사용해 랜덤하게 출력될 수 있도록 한다.
- 문제 풀기 화면에서 지금 선택지의 answerState를 true로 변경해주며 다른 선택지를 누를 수 있게 하는 기능, 즉, 원래 선택됐던 선택지가 비활성화 되는 기능을 만들어주고 반복문을 통해 전체 선택지를 확인하며 answers에 사용자가 선택한 보기 번호를 기록한다.
- 반복문으로 문제의 정답을 체크하여 score를 계산해 화면에 '맞은 문제 / 전체 문제'로 점수를 띄운다.
- WillPopScope 위젯의 onWillPop를 false로 넣어 기출문제 선택지 화면에서 아이폰의 스와이프 기능이나 안드로이드의 백 버튼을 통해 이전에 풀었던 문제의 결과화면으로 되돌아가지 않도록 한다.

3. 실험 및 결과

- 1) 처음 홈 화면으로 한국사능력검정시험의 기본, 심화로 나누어 사용자가 고를 수 있게 한다.



그림 2 - 홈 화면

- 2) 기본, 심화 각각 선택지에서 사용자가 원하는 회차를 선택할 수 있고 문제 풀이 화면으로 넘어간다.



그림 3 - 리스트 화면

3) 기본적으로 퀴즈 데이터는 문제 풀기 화면에 랜덤하게 출력될 수 있도록 하였고 문제 풀기 화면은 기출문제 질문과 4개의 답변 선택지들로 구성하였다.

문제 풀이 화면에서 사용자는 보기를 고르고 다음 문제로 넘어간다. 이때, 사용자가 문제를 풀 시 다른 선택지를 누를 수 있게 하고 답을 하나 선택하기 전에는 다음 문제로 넘어가지 못하게 하였다. 답을 선택하면 보라색으로 어떤 답을 선택하였는지 확인하기 쉽게 구현하였다.



그림 4 - 문제 풀기 화면

4) 사용자가 문제를 다 풀고 나면 결과 화면으로 넘어간다. 이때 문제 결과 화면은 자신이 합격할 수 있는 점수인지 바로 확인할 수 있도록 '맞은 문제 / 전체 문제'로 결과 화면에 나타내었다. 그 다음, 홈으로 돌아가기 버튼을 누르면 기본, 심화를 선택할 수 있는 처음 화면으로 되돌아가서 사용자가 또다른 회차의 문제들을 풀 수 있도록 유도한다.

다시 돌아온 기출문제 선택지 화면에서 아이폰의 스와이프 기능이나 안드로이드의 백 버튼을 통해 이전에 풀었던 문제의 결과화면으로 되돌아가지 않도록 한다.



그림 5 – 문제 결과 화면

4. 결론

한국사능력검정시험 퀴즈앱의 홈 화면, 문제 리스트, 문제 풀기 화면, 문제 결과 화면을 구현하였다. 자투리 시간에 활용하는 것이라 최대한 심플하고 간결하게 구성하였다. 한국사능력검정시험은 기본, 심화로 나누어져 있어 홈 화면에서 선택할 수 있게 하였다. 문제 리스트 화면의 상단에는 기본인지 심화인지 알 수 있게 표시해두었고, '한국사능력검정시험 기본 64' 와 같이 몇 회차 시험지 알 수 있도록 시험지 리스트를 구현하였다.

Django Rest Framework 기반의 API 서버를 개발하여 퀴즈 데이터를 생성 후 문제 풀기 화면에 랜덤하게 출력할 수 있도록 하였다. 문제 풀기 화면은 기출문제 질문과 4개의 답변 선택지들로 구성하였다. 답을 하나 선택하기 전에는 다음 문제로 넘어가지 못하게 하였고 답을 선택하면 보라색으로 어떤 답을 선택하였는지 확인하기 쉽게 구현하였다. 문제 결과 화면은 자신이 합격할 수 있는 점수인지 수험생들이 바로 확인할 수 있도록 '맞은 문제 / 전체 문제'로 결과 화면에 나타내었다.

시간상 시험지 문제들을 모두 추가하지 못하여서 추후에 문제들을 더 추가할 것이다. 또한 추가 기능을 넣지 못했지만 로그인 기능과 타임어택 기능, 폰 문제들을 모아놓는 탭, 틀린 문제들만 모아 오답노트를 할 수 있는 탭, 소셜 로그인(카카오톡) 등을 추가로 구현해 보려 한다. 이후 다른 자격증들도 퀴즈 앱에 넣어 수험생에게 도움이 될 수 있는 앱을 개발할 계획이다.

부록: 소스 코드

깃허브 주소

- <https://github.com/almeng00/quiz-app-project>
- <https://github.com/almeng00/df-quiz-test>

소스코드

```
import 'package:auto_size_text/auto_size_text.dart';
import 'package:flutter/cupertino.dart';
import 'package:flutter/material.dart';
import 'package:flutter_swiper/flutter_swiper.dart';
import 'package:quiz_app_test/model/model_quiz.dart';
import 'package:quiz_app_test/screen/screen_result.dart';
import 'package:quiz_app_test/widget/widget_candidate.dart';

class QuizScreen extends StatefulWidget {
  List<Quiz> quizzes;
  QuizScreen({required this.quizzes});
  @override
  _QuizScreenState createState() => _QuizScreenState();
}

class _QuizScreenState extends State<QuizScreen> {
  List<int> _answers = [-1, -1, -1];
  List<bool> _answerState = [false, false, false, false];
  int _currentIndex = 0;
  SwiperController _controller = SwiperController();
```

```

@override
Widget build(BuildContext context) {
  Size screenSize = MediaQuery.of(context).size;
  double width = screenSize.width;
  double height = screenSize.height;
  return SafeArea(
    child: Scaffold(
      backgroundColor: Colors.deepPurple,
      body: Center(
        child: Container(
          decoration: BoxDecoration(
            borderRadius: BorderRadius.circular(20),
            border: Border.all(color: Colors.deepPurple),
          ),
          width: width * 0.85,
          height: height * 0.5,
          child: Swiper(
            controller: _controller,
            physics: NeverScrollableScrollPhysics(),
            loop: false,
            itemCount: widget.quizzes.length,
            itemBuilder: (BuildContext context, int index) {
              return _buildQuizCard(widget.quizzes[index], width, height);
            },
          ),
        ),
      ),
    ),
  );
}

Widget _buildQuizCard(Quiz quiz, double width, double height) {
  return Container(
    decoration: BoxDecoration(
      borderRadius: BorderRadius.circular(20),
      border: Border.all(color: Colors.white),
      color: Colors.white,
    ),
    child: Column(
      crossAxisAlignment: CrossAxisAlignment.center,
      children: <Widget>[
        Container(
          padding: EdgeInsets.fromLTRB(0, width * 0.024, 0, width * 0.024),
          child: Text(
            'Q' + (_currentIndex + 1).toString() + '.',
            style: TextStyle(
              fontSize: width * 0.06,
              fontWeight: FontWeight.bold,
            ),
          ),
        ),
        Container(
          width: width * 0.8,
          padding: EdgeInsets.only(top: width * 0.012),
          child: AutoSizeText(
            quiz.title,
            textAlign: TextAlign.center,
            maxLines: 2,
            style: TextStyle(
              fontSize: width * 0.048,
              fontWeight: FontWeight.bold,
            ),
          ),
        ),
      ],
    ),
  );
}

```

```

Container(
  padding: EdgeInsets.all(width * 0.024),
  child: Center(
    child: ButtonTheme(
      minWidth: width * 0.5,
      height: height * 0.05,
      shape: RoundedRectangleBorder(
        borderRadius: BorderRadius.circular(10),
      ),
    ),
    child: ElevatedButton(
      child: _currentIndex == widget.quizzes.length - 1
        ? Text('결과보기')
        : Text('다음문제'),
      style: ElevatedButton.styleFrom(
        foregroundColor: Colors.deepPurple, // change background color of button
        backgroundColor: Colors.white, // change text color of button
      ),
      onPressed: _answers[_currentIndex] == -1
        ? null
        : () {
            if (_currentIndex == widget.quizzes.length - 1) {
              Navigator.push(
                context,
                MaterialPageRoute(
                  builder: (context) => ResultScreen(
                    answers: _answers,
                    quizzes: widget.quizzes,
                  ),
                ),
              );
            } else {
              _answerState = [false, false, false, false];
            }
          }
        )
  )
)

```

```

        _answerState = [false, false, false, false];
        _currentIndex += 1;
        _controller.next();
      }
    },
  ),
),
),
),
],
),
);
}

List<Widget> _buildCandidates(double width, Quiz quiz) {
  List<Widget> _children = [];
  for (int i = 0; i < 4; i++) {
    _children.add(
      CandWidget(
        index: i,
        text: quiz.candidates[i],
        width: width,
        answerState: _answerState[i],
        tap: () {
          setState(() {
            for (int j = 0; j < 4; j++) {
              if (j == i) {
                _answerState[j] = true;
                _answers[_currentIndex] = j;
              } else {
                _answerState[j] = false;
              }
            }
          });
        },
      ),
    );
  }
}

```

그림 6 - 문제 풀기 화면

```

import 'package:flutter/material.dart';
import 'package:quiz_app_test/model/model_quiz.dart';
import 'package:quiz_app_test/screen/screen_home.dart';

class ResultScreen extends StatelessWidget {
  List<int> answers;
  List<Quiz> quizzes;
  ResultScreen({required this.answers, required this.quizzes});

  @override
  Widget build(BuildContext context) {
    Size screenSize = MediaQuery.of(context).size;
    double width = screenSize.width;
    double height = screenSize.height;

    int score = 0;
    for (int i = 0; i < quizzes.length; i++) {
      if (quizzes[i].answer == answers[i]) {
        score += 1;
      }
    }
  }
}

```

```

return WillPopScope(
  onWillPop: () async => false,
  child: SafeArea(
    child: Scaffold(
      appBar: AppBar(
        title: Text('한국사능력검정시험'),
        backgroundColor: Colors.deepPurple,
        leading: Container(),
        centerTitle: true,
      ),
      body: Center(
        child: Container(
          decoration: BoxDecoration(
            borderRadius: BorderRadius.circular(20),
            border: Border.all(color: Colors.deepPurple),
            color: Colors.deepPurple,
          ),
          width: width * 0.85,
          height: height * 0.5,
          child: Column(
            children: <Widget>[
              Padding(
                padding: EdgeInsets.only(bottom: width * 0.048),
              ),
              Container(
                decoration: BoxDecoration(
                  borderRadius: BorderRadius.circular(20),
                  border: Border.all(color: Colors.deepPurple),
                  color: Colors.white,
                ),
              ),
            ],
          ),
        ),
      ),
    ),
  ),
);

```



```

width: width * 0.73,
height: height * 0.25,
child: Column(
  children: <Widget>[
    Container(
      padding: EdgeInsets.fromLTRB(
        0, width * 0.048, 0, width * 0.012),
      child: Text(
        '수고하셨습니다!',
        style: TextStyle(
          fontSize: width * 0.055,
          fontWeight: FontWeight.bold,
        ),
      ),
    ),
    Text(
      '당신의 점수는 ',
      style: TextStyle(
        fontSize: width * 0.048,
        fontWeight: FontWeight.bold,
      ),
    ),
    Expanded(
      child: Container(),
    ),
    Text(
      score.toString() + '/' + quizzes.length.toString(),
      style: TextStyle(
        fontSize: width * 0.21,
        fontWeight: FontWeight.bold,
        color: Colors.red,
      ),
    ),
  ],
),
),
Expanded(
  child: Container(),
),
),
Container(
  padding: EdgeInsets.only(bottom: width * 0.048),
  child: ButtonTheme(
    minWidth: width * 0.73,
    height: height * 0.05,
    shape: RoundedRectangleBorder(
      borderRadius: BorderRadius.circular(10),
    ),
  ),
  child: ElevatedButton(
    onPressed: () {
      Navigator.push(context,
        MaterialPageRoute(builder: (context) {
          return HomeScreen();
        }));
    },
    child: Text('홈으로 돌아가기'),
    style: ElevatedButton.styleFrom(
      foregroundColor: Colors.white, // change background color of button
      backgroundColor: Colors.black, // change text color of button
    ),
  ),
),
),

```

그림 7 - 문제 결과 화면