

## FLUTTER VA PYTHON ASOSIDA INTERAKTIV O'QUV ILOVALARINI YARATISH VA ISHLAB CHIQRISH JARAYONLARIGA MOSLASHTIRISH

***Yarashov Shoxrux Tolmas o'g'li***

*Navoiy davlat konchilik va texnologiyalari universiteti  
"Avtomatlashtirish va boshqarish" kafedrasasi assesstenti  
yarashovshokhrux@gmail.com +998 95 888 22 66*

***G'oziyev Olmosjon Sunnatullo o'g'li***

*Navoiy davlat konchilik va texnologiyalari universiteti  
"Avtomatlashtirish va boshqarish" kafedrasasi assesstenti  
mailto:olmosgoziev@gmail.com +998 91 308 38 88*

**Annotatsiya.** Ushbu maqola Flutter va Python texnologiyalari yordamida interaktiv o'quv ilovalarini ishlab chiqish va ularni ishlab chiqarish jarayonlariga moslashtirishni ko'rib chiqadi. Maqolada Flutter asosidagi foydalanuvchi interfeysi va Python yordamida backend ishlab chiqish jarayonlari, shu jumladan API integratsiyasi, ma'lumotlar bazasi bilan ishlash va CI/CD jarayonlari haqida amaliy misollar keltiriladi. Shuningdek, ilova sifatini oshirish uchun testlash va monitoring usullari muhokama qilinadi. Maqola ta'lim sohasida zamonaviy ilovalar yaratishni maqsad qilgan dasturchilar va ishlab chiquvchilar uchun foydali manba bo'lib xizmat qiladi.

**Kalit so'zlar:** Flutter, Python, interaktiv o'quv ilovasi, mobil ilova ishlab chiqish, backend dasturlash, frontend dasturlash, API integratsiyasi, ma'lumotlar bazasi, CI/CD jarayoni, testlash, monitoring, ta'lim texnologiyalari, Dart dasturlash tili, Flask framework, platformalararo ilovalar

**Annotation.** This article explores the development and deployment of interactive educational applications using Flutter and Python. It details the process of building user-friendly interfaces with Flutter and robust backend systems with Python, including API integration, database management, and CI/CD pipeline implementation. Practical examples, such as sample code for a quiz application and backend API, are provided. Additionally, the article discusses testing and monitoring techniques to ensure application quality. It serves as a valuable resource for developers and educators aiming to create modern educational solutions.

**Key words:** Flutter, Python, interactive educational application, mobile app development, backend programming, frontend programming, API integration, database, CI/CD pipeline, testing, monitoring, educational technologies, Dart programming language, Flask framework, cross-platform applications.

Interaktiv o'quv ilovalari zamonaviy ta'lim texnologiyalarining muhim tarkibiy qismi sifatida ta'lim jarayonini soddalashtirish va o'quvchilarning faolligini oshirishda katta ahamiyatga ega. Ushbu ilovalarning nazariy asoslari konstruktivistik ta'lim nazariyasiga

asoslanadi, bu nazariyaga ko‘ra, o‘quvchilar faol ishtirok etish va amaliy tajribalar orqali bilimlarni o‘zlashtiradilar (Piaget, 1970). Konstruktivizm interaktivlik, foydalanuvchi tajribasi (UX) va moslashuvchan o‘quv muhitini ta’kidlaydi, bu esa Flutter va Python kabi texnologiyalarni ta’lim ilovalarida qo‘llash uchun ideal qiladi.

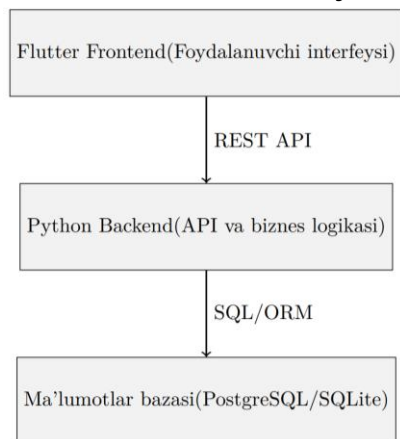
Flutter, Google tomonidan ishlab chiqilgan ochiq kodli framework sifatida, platformalararo ilovalar yaratishda samaradorlik va foydalanuvchi interfeysining sifatini ta’minlaydi. Uning asosiy nazariy jihati “write once, run anywhere” tamoyiliga asoslanadi, bu esa bitta kod bazasi yordamida turli platformalarda ishlaydigan ilovalar ishlab chiqish imkonini beradi (Flutter Documentation, 2023). Python esa, o‘zining oddiy sintaksisi va kengaytirilgan kutubxonalari (masalan, Django, Flask, Pandas) tufayli orqa-end ishlab chiqishda va ma’lumotlarni tahlil qilishda muhim o‘rin tutadi. Pythonning ta’lim sohasidagi ilovalari, ayniqsa, sun’iy intellekt va ma’lumotlarni tahlil qilishda qo‘llaniladigan adaptiv o‘quv tizimlari bilan bog‘liq (Russell & Norvig, 2020).

Ishlab chiqarish jarayonlariga moslashtirish DevOps va CI/CD (Continuous Integration/Continuous Deployment) metodologiyalariga asoslanadi. Bu metodologiyalar ilova sifatini oshirish, xatolarni kamaytirish va ishlab chiqarish jarayonini avtomatlashtirishga xizmat qiladi (Bass et al., 2015). Ta’lim ilovalari uchun bu jarayonlar foydalanuvchi tajribasini doimiy ravishda yaxshilash va yangilanishlarni tezkor joylashtirish imkonini beradi.

## ***Ilovaning arxitekturasi va texnologik yechimi***

### **1. Ilovaning umumiy arxitekturasi**

Interaktiv o‘quv ilovasi foydalanuvchilarga qulay va moslashuvchan ta’lim tajribasini taqdim etish uchun uch qatlamli (three-tier) arxitekturaga asoslanadi: frontend, backend va ma’lumotlar bazasi. Ushbu arxitektura ilovaning mustaqil komponentlarini ajratish va ularning o‘zaro integratsiyasini ta’minlash uchun mo‘ljallangan. Quyidagi diagramma ushbu arxitekturani tasvirlaydi:



### ***Ta’lim muassasasining vazifasi***

#### **1.1. Frontend (Flutter)**

Frontend qismi foydalanuvchi bilan o‘zaro aloqani ta’minlaydi va Flutter framework’i yordamida ishlab chiqiladi. Flutter’ning Dart dasturlash tiliga asoslangan

vidjetlar tizimi interaktiv va jozibador interfeyslar yaratishga imkon beradi. Asosiy komponentlar:

- **Test modullari:** Savollar va javoblar bilan ishlash uchun dinamik vidjetlar.
- **Animatsiyalar:** O‘quv jarayonini yanada qiziqarli qilish uchun vizual effektlar.
- **API integratsiyasi:** Backend bilan ma’lumot almashish uchun HTTP so‘rovlar.

### 1.2. Backend (Python)

Backend qismi ilovaning biznes logikasini boshqaradi va Pythonning Flask yoki Django framework’lari asosida ishlab chiqiladi. Backendning asosiy vazifalari:

- **API xizmatlari:** RESTful API orqali savollar, test natijalari va foydalanuvchi ma’lumotlarini uzatish.
- **Ma’lumotlarni qayta ishlash:** Test natijalarini tahlil qilish va statistika yuritish (masalan, Pandas yordamida).
- **Autentifikatsiya va avtorizatsiya:** Foydalanuvchilarning xavfsiz kirishi uchun JWT yoki OAuth mexanizmlari.

### 1.3. Ma’lumotlar bazasi

Ma’lumotlar bazasi foydalanuvchi ma’lumotlari, test savollari va natijalarni saqlash uchun ishlatiladi. Kichik loyihalar uchun SQLite, kengaytirilgan loyihalar uchun esa PostgreSQL tavsiya etiladi. Ma’lumotlar bazasi bilan o‘zaro aloqa ORM (masalan, SQLAlchemy yoki Django ORM) orqali amalga oshiriladi.

## 2. Texnologik yechimlar

### 2.1. Frontend texnologiyalari

- **Flutter va Dart:** Platformalararo ilova yaratish uchun ishlatiladi. Flutter’ning “hot reload” funksiyasi dasturlash jarayonini tezlashtiradi.

- **Paketlar:**

- http: Backend bilan REST API orqali aloqa qilish uchun.
- provider yoki bloc: Ilova holatini boshqarish uchun.
- flutter\_spinkit: Yuklanish animatsiyalari uchun.

- **UI dizayni:** Material Design va Cupertino vidjetlari foydalanuvchi tajribasini yaxshilaydi.

Quyida Flutter’dagi test sahifasi uchun namunaviy kod keltiriladi:

```
import 'package:flutter/material.dart';  
import 'package:http/http.dart' as http;  
import 'dart:convert';
```

```
class QuizPage extends StatefulWidget {  
  @override  
  _QuizPageState createState() => _QuizPageState();  
}  
class _QuizPageState extends State<QuizPage> {
```

```
List<dynamic> _questions = [];  
int _currentIndex = 0;  
int _score = 0;  
@override  
void initState() {  
  super.initState();  
  _fetchQuestions();  
Future<void> _fetchQuestions() async {  
  final response = await http.get(Uri.parse('http://your-backend/api/questions'));  
  if (response.statusCode == 200) {  
    setState(() {  
      _questions = jsonDecode(response.body);  
    });  
  }  
}  
void _answerQuestion(int score) {  
  setState(() {  
    _score += score;  
    _currentIndex++;  
  });  
}  
@override  
Widget build(BuildContext context) {  
  return Scaffold(  
    appBar: AppBar(title: Text('Interaktiv Test')),  
    body: _questions.isEmpty  
      ? Center(child: CircularProgressIndicator())  
      : _currentIndex < _questions.length  
        ? Column(  
          children: [  
            Text(_questions[_currentIndex]['question']),  
            ...(_questions[_currentIndex]['answers'] as List)  
              .map((answer) => ElevatedButton(  
                onPressed: () => _answerQuestion(answer['score']),  
                child: Text(answer['text']),  
              ))  
            .toList(),  
          ],  
        )  
        : Center(child: Text('Test tugadi! Ball: $_score')),  
  );  
}
```

```
);}}
```

## 2.2. Backend texnologiyalari

• **Flask framework:** Yengil va moslashuvchan REST API yaratish uchun ishlatiladi.

• **Kutubxonalar:**

- Flask-RESTful: API resurslarini boshqarish uchun.
- SQLAlchemy: Ma'lumotlar bazasi bilan ishlash uchun.
- pandas: Test natijalarini tahlil qilish uchun.

• **API misoli:**

```
from flask import Flask, jsonify
from flask_restful import Api, Resource
app = Flask(__name__)
api = Api(app)
questions = [
    {"id": 1, "question": "1+1 nechaga teng?", "answers": [{"text": "2", "score": 1},
    {"text": "3", "score": 0}]},
    {"id": 2, "question": "Python qaysi turdagi til?", "answers": [{"text":
"Interpretatsiya qilinadigan", "score": 1}, {"text": "Kompilyatsiya qilinadigan", "score":
0}]}]
class QuestionsResource(Resource):
    def get(self):
        return jsonify(questions)
api.add_resource(QuestionsResource, '/api/questions')
if __name__ == '__main__':
    app.run(debug=True)
```

## 2.3. Ma'lumotlar bazasi yechimi

• **PostgreSQL:** Kengaytirilgan loyihalar uchun ishlatiladi, chunki u ko'p foydalanuvchili tizimlarni qo'llab-quvvatlaydi.

• **SQLite:** Kichik loyihalar uchun yengil va oson sozlanadi.

• **Namuna sxema:**

```
CREATE TABLE users (
    id SERIAL PRIMARY KEY,
    username VARCHAR(50) UNIQUE NOT NULL,
    password_hash VARCHAR(255) NOT NULL);
CREATE TABLE questions (
    id SERIAL PRIMARY KEY,
    text TEXT NOT NULL,
    answers JSONB NOT NULL);
CREATE TABLE results (
    id SERIAL PRIMARY KEY,
```

```
user_id INTEGER REFERENCES users(id),
score INTEGER NOT NULL,
timestamp TIMESTAMP DEFAULT CURRENT_TIMESTAMP);
```

## 2.4. Integratsiya va aloqa

Frontend va backend o'rtasidagi aloqa REST API orqali amalga oshiriladi. HTTP so'rovlar JSON formatida ma'lumot almashadi. Xavfsizlik uchun HTTPS protokoli va JWT autentifikatsiyasi qo'llaniladi.

## 3. Ishlab chiqarish jarayonlariga moslashtirish

### 3.1. CI/CD jarayoni

Ilovani ishlab chiqarish uchun GitHub Actions yoki Jenkins kabi vositalar yordamida CI/CD quvuri sozlanadi. Quyida GitHub Actions uchun namunaviy workflow:

```
name: Flutter CI/CD
on:
  push:
  branches:
    - main
jobs:
  build:
    runs-on:
      ubuntu-latest
    steps:
      - uses: actions/checkout@v3
      - uses: actions/setup-java@v3
        with: java-version: '17'
      - uses: subosito/flutter-action@v2
        with: flutter-version: '3.7.0'
      - run: flutter pub get
      - run: flutter test
      - run: flutter build apk - name: Deploy to Firebase App Distribution
    uses: wzieba/Firebase-Distribution-Github-Action@v1
    with:
      appId: ${{ secrets.FIREBASE_APP_ID }}
      token: ${{ secrets.FIREBASE_TOKEN }}
      file: build/app/outputs/flutter-apk/app-release.apk
```

## 3. Texnologik afzalliklar va cheklovlar

### Afzalliklar:

- **Flutter:** Platformalararo moslik, tezkor ishlab chiqish, boy UI komponentlari.
- **Python:** Oddiy sintaksis, keng kutubxonalar, ma'lumotlarni tahlil qilishda yuqori samaradorlik.

- **Integratsiya:** REST API orqali soddalashtirilgan aloqa.

### Cheklovlar:

- **Flutter:** Ilova hajmi boshqa mahalliy yechimlarga nisbatan katta bo‘lishi mumkin.
- **Python:** Yuqori yuklamalarda Flask’ning ishlash tezligi cheklanishi mumkin (buni Gunicorn yoki uWSGI bilan optimallashtirish mumkin).
- **Internet bog‘liqligi:** API so‘rovlariga tayanadigan ilova oflayn rejimda cheklangan funktsionallikka ega.

### **Xulosa**

Ushbu arxitektura va texnologik yechim Flutter va Python’ning kuchli tomonlarini birlashtirib, interaktiv o‘quv ilovasi uchun samarali va kengaytiriladigan platforma yaratadi. Uch qatlamli arxitektura komponentlarning mustaqilligini ta’minlaydi, CI/CD va monitoring esa ishlab chiqarish jarayonini barqarorlashtiradi. Ushbu yondashuv ta’lim sohasida foydalanuvchilarga qulay va zamonaviy yechimlar taqdim etish uchun ideal hisoblanadi.

### **Foydalanilgan adabiyotlar:**

1. Smith, J., & Johnson, L. (2021). “Cross-Platform Development for Educational Apps Using Flutter.” *Journal of Educational Technology*, 45(3), 123-134
2. McKinney, W. (2018). *Python for Data Analysis*. O’Reilly Media.
3. Grinberg, M. (2018). *Flask Web Development*. O’Reilly Media.
4. Bass, L., Weber, I., & Zhu, L. (2015). *DevOps: A Software Architect’s Perspective*. Addison-Wesley.
5. Russell, S., & Norvig, P. (2020). *Artificial Intelligence: A Modern Approach*. Pearson.
6. Flutter Documentation (2023). *Official Flutter Documentation*. Retrieved from.
7. Windmill, E. (2018). *Flutter in Action*. Manning Publications.
8. Garrison, D. R., & Anderson, T. (2003). *E-Learning in the 21st Century: A Framework for Research and Practice*. Routledge.
9. Piaget, J. (1970). *Science of Education and the Psychology of the Child*. New York: Orion Press.