

Compositions of Ukranian Jazz/Classical Composer Nikolai Kapustin (November 22, 1937 - July 2, 2020): A Database, API, and WebUI

Project Description

In the realm of contemporary classical and jazz music, few composers have bridged the gap as seamlessly as Nikolai Kapustin. A virtuoso pianist and prolific composer, Kapustin's works are a testament to the harmonious blend of classical structures with jazz improvisation. Yet, despite his monumental contributions, his works remain underrepresented in the digital world.

This project aims to create a comprehensive digital repository of Kapustin's musical works. From orchestral masterpieces to intimate solo piano pieces, our API offers a deep dive into the world of Kapustin. Whether you're a researcher, a musician looking for sheet music, or simply a fan, our platform provides detailed information, categorized meticulously by type, opus number, movements, and more. Accompanied by YouTube links for each work, users can both explore and experience the magic of Kapustin's compositions.

Tools & Technology

- 1. Database:
 - 1. SQLite: A lightweight, serverless, self-contained SQL database engine. It's perfect for projects where you want a simple relational database without the overhead of setting up a server.
- 2. Backend/API:
 - 1. Back4App: A platform that provides backend services to help developers create apps faster. It's built on top of Parse, an open-source backend framework. With Back4App, we can easily set up a database, create API endpoints, and manage user authentication.
- 3. Frontend:
 - 1. HTML/CSS/JavaScript: Basic web technologies to create the single-page application that interacts with our API.
- 4. API Testing:
 - 1. Postman: A popular tool for testing API endpoints. It allows you to send requests to your API and view responses, making development and debugging easier.
- 5. Version Control:
 - 1. Git & GitHub: To manage versions of your code, collaborate with your partner, and maintain a history of changes.

Project Timeline (7 days)

- 1. Design & Planning (1 day):
 - 1. Finalize the database schema
 - 2. Plan out the API endpoints
 - 3. Scetch a basic wireframe for the frontend
- 2. Setting up the Database & Back4App (1 day):
 - 1. Initialize the SQLite database
 - 2. Define tables and relationships in Back4App
 - 3. Populate the database with some initial data for testing
- 3. Developing the API (2 days):
 - 1. Set up basic CRUD operations for the `Opus` table
 - 2. Implement search and filter endpoints
 - 3. Set up user authentication
 - 4. Test all endpoints using Postman
- 4. Developing the Frontend (2 days):
 - 1. Set up a basic web page layout
 - 2. Implement API calls from the frontend
 - 3. Add user authentication flow (login/register)
 - 4. Implement data display, search, and filter functionalities.
- 5. Testing & Debugging (0.5 day):
 - 1. Thoroughly test the entire application flow
 - 2. Debug any issues that arise
- 6. Stretch: Documentation & Final Touches (0.5 day):
 - 1. Write API documentation
 - 2. Add comments to the code
 - 3. Make any final adjustments to the frontend design or functionality