

# Piaðo Progression

*Nasywaa Anggun Athiefah - 18222021*

*II3160 - Teknologi Sistem Terintegrasi*

---

Piano Progression adalah platform yang dirancang untuk mendukung pembelajaran dan latihan piano. Website ini membantu siswa dan pengajar mengatur jadwal latihan, memantau progres, dan mengakses sumber belajar interaktif. Menggabungkan teknologi modern dengan metode pembelajaran piano tradisional, platform ini cocok untuk semua tingkat pembelajaran.

## Daftar Isi

Daftar Isi.....	1
A. Website Overview.....	2
B. Business Capability Map.....	3
Strategic.....	3
Core.....	3
Support.....	3
C. Value Stream.....	4
D. Sub-Domain.....	4
Core (C).....	4
Support (S).....	4
General (G).....	4
E. Tech Stack.....	5
Frontend.....	5
Backend.....	5
Deployment.....	5
F. Fungsionalitas Fitur.....	6
User Authentication.....	6
Landing Page.....	6
ABRSM Curriculum.....	6
Practice Pathway.....	6
Progress Visionary.....	6
Flashcard.....	7
Feedback and Evaluation.....	7
G. Integrasi Service.....	7
H. Kesimpulan.....	9

## A. Website Overview

<b>Link deployment frontend</b>	api.pianoprogression.web.id
<b>Link deployment backend</b>	pianoprogression.web.id
<b>Link github frontend</b>	github.com/nasywaanaa/PianoProgressionFE.git
<b>Link github frontend</b>	github.com/nasywaanaa/PianoProgressionBE.git

**Piano Progression** adalah website pembelajaran piano yang dirancang khusus untuk mendukung pengguna yang mengikuti kurikulum **ABRSM (Associated Board of the Royal Schools of Music)** dari Grade 1 hingga Grade 8. Website tersebut digunakan untuk memberikan cara belajar piano yang terstruktur dan memudahkan seseorang yang ingin mengikuti ujian ABRSM dengan berbagai fitur yang nantinya dapat memberikan progress evaluation.

### 1. Tujuan :

- a. Merencanakan pembelajaran piano sesuai dengan kurikulum ABRSM
- b. Mengatur jadwal pembelajaran piano dan memberikan penjadwalan otomatis dari task masukan user
- c. Melacak progres belajar (*tracker*)
- d. Melatih untuk theory exam dengan adanya flash card
- e. Memberikan *evaluation report* atas performa user

### 2. Fitur :

#### a. User Authentication

- i. Login dan registrasi dengan opsi menggunakan Google

#### b. Landing Page

- i. Menjelaskan manfaat dan tujuan platform
- ii. Navigasi ke fitur utama : ABRSM Navigator, Practice Pathway, Progress Visionary, dan Interactive Essentials
- iii. Testimoni user yang pernah menggunakannya

#### c. ABRSM Navigator

- i. Panduan untuk pelacakan progres sesuai kurikulum ABRSM
- ii. Informasi tentang ujian teori dan praktik, termasuk aural test, sight reading, scales, arpeggios, dan performance pieces.

**d. Practice Pathway**

- i. Kalender untuk mengelola jadwal latihan
- ii. Fitur tambah jadwal dengan detail seperti lagu, penulis, grade, dan catatan tambahan
- iii. Evaluasi performa dengan skor dan umpan balik mendetail.

**e. Progress Visionary**

- i. Visualisasi data progres dalam tabel yang menampilkan informasi lagu, grade, tanggal, dan status
- ii. Statistik performa bulanan dalam bentuk grafik

**f. Flashcard**

- i. Flashcard untuk mempelajari teori musik seperti notasi, tanda kunci, dan skala
- ii. Quiz dengan pertanyaan dan penjelasan tambahan

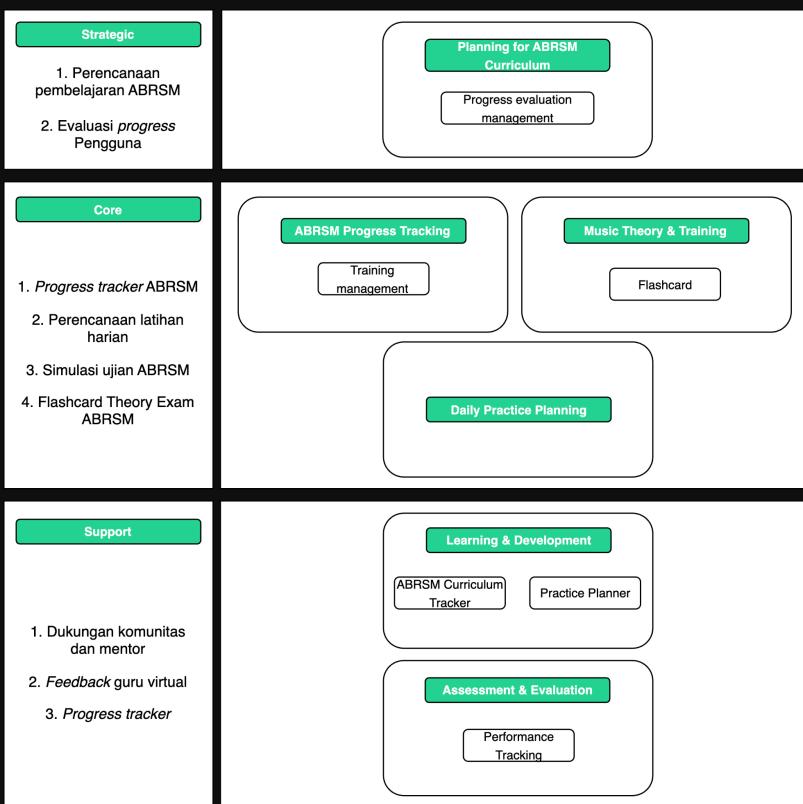
**g. Feedback and Evaluation**

- i. Umpan balik otomatis untuk latihan praktik
- ii. Evaluasi performa dengan saran perbaikan yang spesifik

### 3. Integrasi Eksternal :

Piano progression mengintegrasikan service external yaitu **Taskly (18222021 - Jihan Aurelia)** untuk pengelolaan jadwal tugas. API eksternal digunakan untuk menjadwalkan tugas dengan memberikan api\_key sehingga keamanan service tetap terjaga.

## B.Business Capability Map



### Strategic

Perencanaan pembelajaran ABRSM membantu pengguna merancang strategi belajar yang sesuai dengan kurikulum, sementara evaluasi progres pengguna memastikan kemajuan mereka dipantau dan dinilai secara berkala untuk tetap berada di jalur yang benar.

### Core

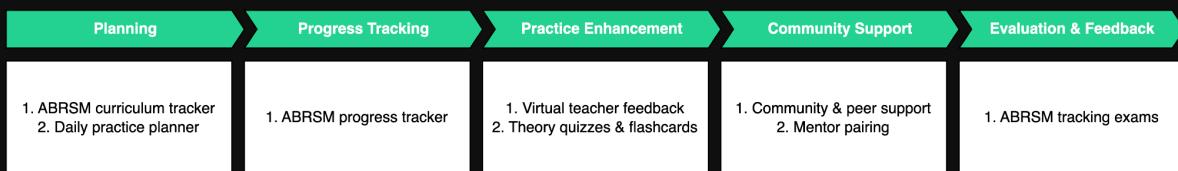
#### Progress tracker

ABRSM melacak perkembangan pengguna terhadap target yang ditetapkan. Perencanaan latihan harian menyediakan jadwal latihan terstruktur, sedangkan simulasi ujian ABRSM membantu pengguna mempersiapkan diri menghadapi ujian. Flashcard theory exam ABRSM adalah media interaktif untuk mempelajari teori musik dengan lebih mudah.

### Support

Dukungan komunitas dan mentor memperkuat pengalaman belajar dengan akses ke komunitas pembelajaran dan bimbingan dari mentor. Feedback guru virtual memberikan masukan otomatis berbasis AI untuk meningkatkan performa, didukung oleh progress tracker yang memvisualisasikan perkembangan pengguna secara keseluruhan.

## C. Value Stream



**Planning** mencakup ABRSM curriculum tracker dan daily practice planner untuk membantu pengguna merencanakan pembelajaran dan latihan harian secara terstruktur. **Progress Tracking** dapat membuat pengguna memantau perkembangan mereka melalui ABRSM progress tracker. **Practice Enhancement** menyediakan feedback dari guru virtual serta latihan teori menggunakan flashcard. **Community Support** menawarkan dukungan melalui komunitas belajar, kolaborasi dengan rekan, dan pendampingan mentor. **Evaluation & Feedback** untuk mengevaluasi progress.

## D. Sub-Domain



### Core (C)

Fitur utama mencakup Pelacak Kurikulum ABRSM untuk memantau kemajuan, Perencana Latihan Harian untuk jadwal latihan, dan Visualisasi Progress untuk evaluasi progress.

### Support (S)

Fitur pendukung mencakup Latihan Sight-Reading & Aural, Kuis & Kartu Flash Teori untuk teori musik interaktif, dan Dukungan Komunitas untuk kolaborasi dengan rekan dan mentor.

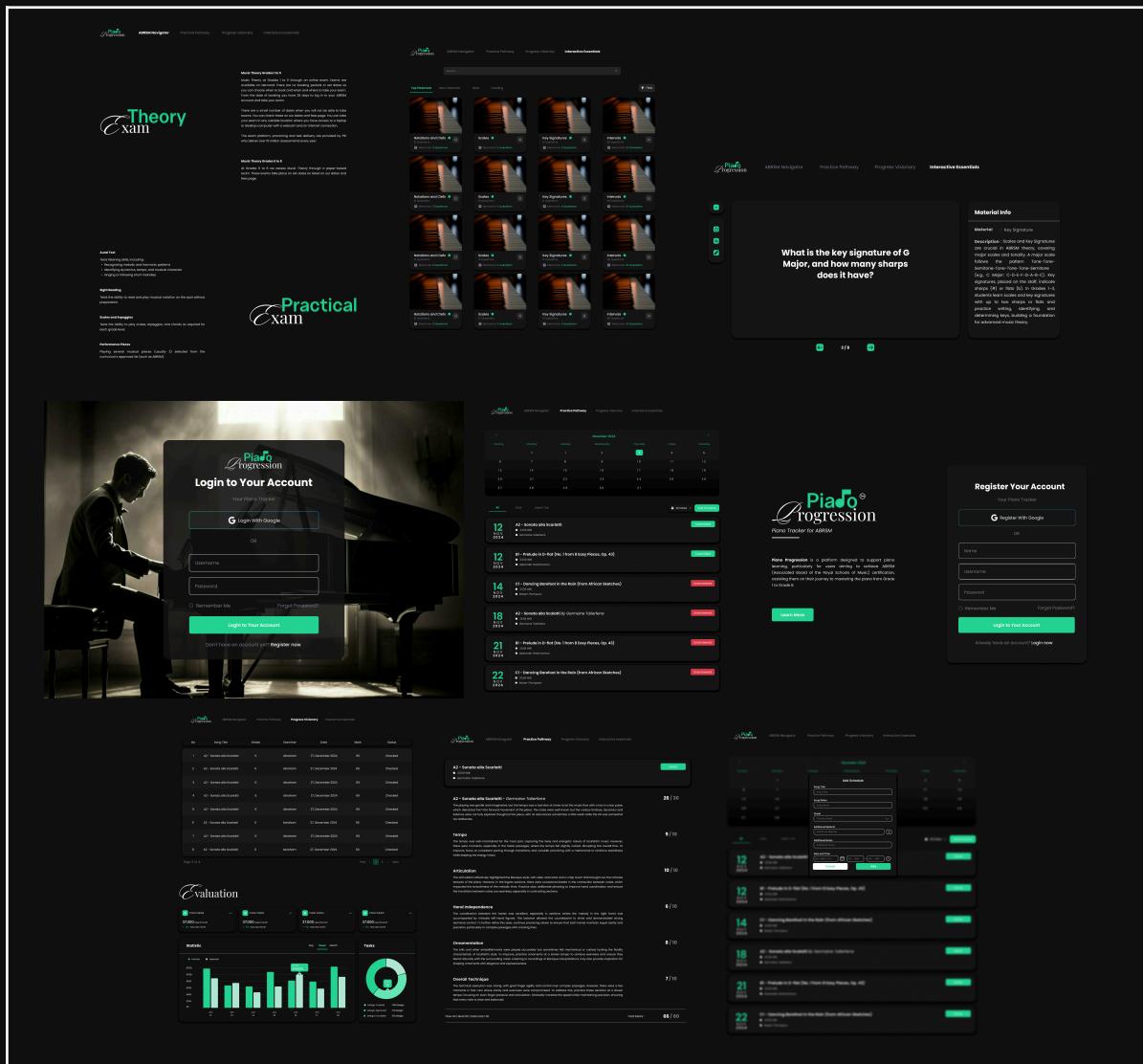
### General (G)

Berfokus pada keamanan, mencakup Sistem Notifikasi untuk pengingat, Autentikasi dan Manajemen Pengguna, serta Cloud Infrastructure & Data Security untuk perlindungan data berbasis cloud.

## E. Tech Stack

Tech	Description (Alasan)
UI/UX	1. Aplikasi : a. Figma
Frontend	2. Framework : b. React.js : reusable dan dinamis i. Vite ii. Tailwind iii. React router dom iv. Protected route v. Axios
Backend	1. Framework : a. Express.js : server backend fleksibel b. Node.js 2. Database : a. MongoDB : database NoSQL dan scalable 3. Authentication : a. JWT (JSON Web Token) 4. Middleware : a. Body-Parser b. CORS c. Bcrypt
Deployment	1. Docker 2. Vercel (frontend) : deployment untuk frontend 3. VPS (backend) : deployment untuk backend dan dapat membuat domain

## 1. UI/UX



## 2. Frontend:

```
/** @type {import('tailwindcss').Config} */
export default {
  content: [
    "./index.html",
    "./src/**/*.{js,ts,jsx,tsx}",
  ],
  theme: {
    extend: {},
  },
  plugins: [],
}

import { defineConfig } from 'vite'
import react from '@vitejs/plugin-react'
// https://vite.dev/config/
```

```

export default defineConfig({
  plugins: [react()],
})

{
  "name": "pianoprogressionontst",
  "private": true,
  "version": "0.0.0",
  "type": "module",
  "scripts": {
    "dev": "vite",
    "build": "vite build",
    "lint": "eslint .",
    "preview": "vite preview"
  },
  "dependencies": {
    "axios": "^1.7.9",
    "prop-types": "^15.8.1",
    "react": "^18.3.1",
    "react-dom": "^18.3.1",
    "react-router-dom": "^7.1.1"
  },
  "devDependencies": {
    "@eslint/js": "^9.17.0",
    "@types/react": "^18.3.18",
    "@types/react-dom": "^18.3.5",
    "@vitejs/plugin-react": "^4.3.4",
    "autoprefixer": "^10.4.20",
    "eslint": "^9.17.0",
    "eslint-plugin-react": "^7.37.2",
    "eslint-plugin-react-hooks": "^5.0.0",
    "eslint-plugin-react-refresh": "^0.4.16",
    "globals": "^15.14.0",
    "postcss": "^8.4.49",
    "rollup": "^4.30.1",
    "tailwindcss": "^3.4.17",
    "vite": "^6.0.7"
  }
}

```

### 3. Backend:

```

{
  "name": "pianoprogressionbackend",
  "version": "1.0.0",
  "main": "index.js",
  "scripts": {
    "test": "echo \\\"Error: no test specified\\\" && exit 1",
    "start": "node server.js"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "dependencies": {
    "axios": "^1.7.9",
    "bcrypt": "^5.1.1",
    "cors": "^2.8.5",
  }
}

```

```

    "dotenv": "^16.4.7",
    "express": "^4.21.2",
    "jsonwebtoken": "^9.0.2",
    "mongodb": "^6.12.0",
    "mongoose": "^8.9.3"
  },
  "description": ""
}

```

#### 4. Algoritma core business (pseudocode) :

```

POST /schedule/generate/:id
  Input: User ID (from params), Token (for verification)
  Output: Generated schedule or existing schedule, or an error
message

BEGIN
  VERIFY user token using `verifyToken` middleware

  TRY
    EXTRACT `id` from URL params
    LOG "Generating schedule for User ID: <id>"

    GET current date

    FETCH unanswered tasks for the user
    IF no tasks available THEN
      RETURN response with status 400, message "No tasks
available for scheduling"

    SORT tasks by their deadlines in ascending order

    FETCH existing schedules for the user

    CONVERT existing schedule task IDs into a set
    CONVERT current task IDs into a set

    IF size of both sets is equal AND every current task ID
exists in the existing task IDs THEN
      RETURN response with status 200, message "Schedule is
up-to-date. No changes needed.", data: existing schedules

    LOG "Regenerating schedules..."
    DELETE all existing schedules for the user

    INITIALIZE `newSchedules` as an empty list
    SET `currentDay` to today's date

    FOR each task in sorted tasks DO
      ADD a new schedule object to `newSchedules`:
        taskId: task._id,
        userId: user ID from params,

```

```

    date: currentDay
    INCREMENT `currentDay` by 1 day

    SAVE `newSchedules` to the database
    RETURN response with status 201, message "Schedule
generated successfully", data: created schedules

    CATCH any errors
        LOG the error
        RETURN response with status 500, message "Failed to
generate schedule", error message
    END

```

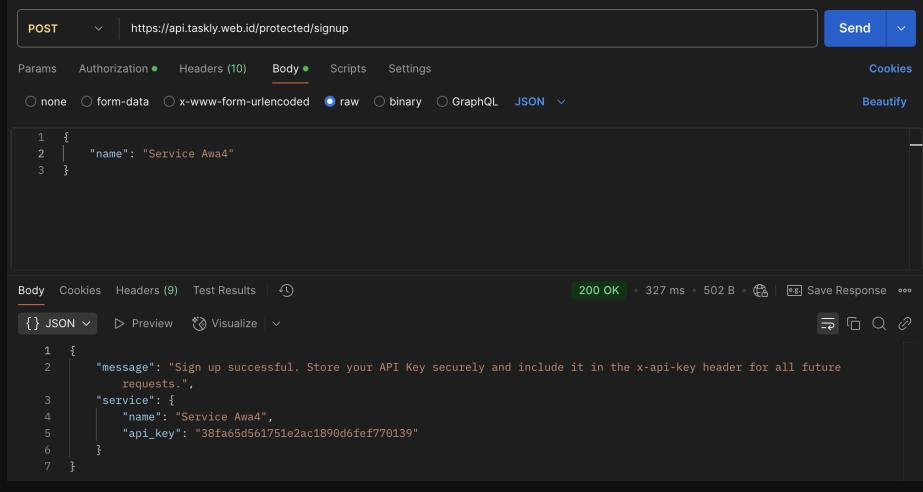
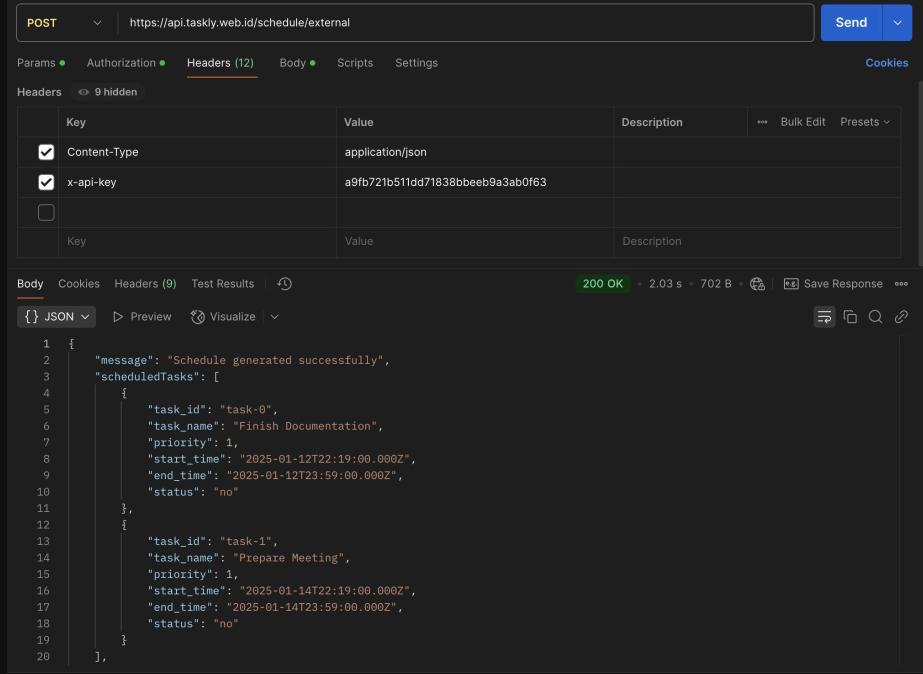
## F. Fungsionalitas Fitur

Fitur	Description (Penjelasan)
User Authentication	<ol style="list-style-type: none"> <li>Halaman login dan registrasi dengan opsi login menggunakan Google.</li> <li>Fitur "Remember Me" untuk menyimpan sesi pengguna.</li> <li>Dukungan untuk pengaturan ulang kata sandi.</li> </ol>
Landing Page	<ol style="list-style-type: none"> <li>Penjelasan tentang platform Piano Progression dan manfaatnya.</li> <li>Navigasi mudah ke berbagai fitur utama: ABRSM Navigator, Practice Pathway, Progress Visionary, dan Interactive Essentials.</li> <li>Testimoni pengguna untuk meningkatkan kredibilitas platform.</li> </ol>
ABRSM Curriculum	<ol style="list-style-type: none"> <li>Panduan lengkap untuk pelacakan progres berdasarkan kurikulum ABRSM.</li> <li>Informasi mendalam tentang ujian teori dan praktik, termasuk aural test, sight reading, scales, arpeggios, dan performance pieces.</li> </ol>
Practice Pathway	<ol style="list-style-type: none"> <li>Kalender interaktif untuk menjadwalkan dan mengelola latihan piano.</li> </ol>

	<ol style="list-style-type: none"> <li>2. Fitur tambah jadwal dengan detail seperti judul lagu, penulis, grade, dan catatan tambahan.</li> <li>3. Penilaian performa pengguna dengan skor dan umpan balik mendetail tentang tempo, artikulasi, teknik, dan lain-lain.</li> </ol>
Progress Visionary	<ol style="list-style-type: none"> <li>1. Visualisasi data progres pengguna dalam tabel yang menampilkan informasi lagu, grade, penguji, tanggal, skor, dan status.</li> <li>2. Statistik pengguna seperti jumlah pengunjung dan grafik evaluasi performa dalam bentuk mingguan dan bulanan.</li> </ol>
Flashcard	<ol style="list-style-type: none"> <li>1. Koleksi flashcard interaktif untuk mempelajari teori musik, termasuk notasi, skala, interval, dan tanda kunci.</li> <li>2. Quiz interaktif dengan pertanyaan mendetail tentang teori musik dan fitur "Material Info" untuk penjelasan tambahan.</li> </ol>
Feedback and Evaluation	<ol style="list-style-type: none"> <li>1. Evaluasi skor total untuk setiap latihan dengan saran perbaikan.</li> </ol>

## G. Integrasi Service

<b>Nama Pemilik Service</b>	Jihan Aurelia (18222001)
<b>Nama Service</b>	Taskly
<b>Link Deployment</b>	taskly.web.id
<b>Konsep</b>	<ol style="list-style-type: none"> <li>1. Sign up terlebih dahulu ke service yang ingin diimplementasikan</li> </ol>

	<ol style="list-style-type: none"> <li>2. Mengambil API_KEY yang diberikan</li> <li>3. Menambahkan schedule dengan service yang ingin diimplementasikan dan dimasukkan API_KEY nya</li> </ol>
Bukti	 <p>The screenshot shows a POST request to <a href="https://api.taskly.web.id/protected/signup">https://api.taskly.web.id/protected/signup</a>. The body contains a JSON object with a single key-value pair: "name": "Service Awa4". The response status is 200 OK, and the response body is a JSON object containing a message, a service object with a name and api_key, and a scheduled_tasks array.</p> <pre> 1 { 2   "name": "Service Awa4" 3 }         </pre> <pre> 1 { 2   "message": "Sign up successful. Store your API Key securely and include it in the x-api-key header for all future requests.", 3   "service": { 4     "name": "Service Awa4", 5     "api_key": "38fa65d561751e2ac1890d6fef770139" 6   } 7 }         </pre>  <p>The screenshot shows a POST request to <a href="https://api.taskly.web.id/schedule/external">https://api.taskly.web.id/schedule/external</a>. The headers section shows Content-Type set to application/json and x-api-key set to a9fb721b511dd71838bbeeb9a3ab0f63. The body contains a JSON object with a message and a scheduledTasks array. The response status is 200 OK, and the response body is a JSON object with a message and a scheduledTasks array.</p> <pre> 1 { 2   "message": "Schedule generated successfully", 3   "scheduledTasks": [ 4     { 5       "task_id": "task-0", 6       "task_name": "Finish Documentation", 7       "priority": 1, 8       "start_time": "2025-01-12T22:19:00.000Z", 9       "end_time": "2025-01-12T23:59:00.000Z", 10      "status": "no" 11    }, 12    { 13      "task_id": "task-1", 14      "task_name": "Prepare Meeting", 15      "priority": 1, 16      "start_time": "2025-01-14T22:19:00.000Z", 17      "end_time": "2025-01-14T23:59:00.000Z", 18      "status": "no" 19    } 20  ], 21 }         </pre>
Endpoint Service Taskly	<p>Schedule  Generate Schedule  URL: /schedule/generate  Method: POST  Headers:</p> <p>Authorization: Bearer &lt;JWT_TOKEN&gt;</p> <p>Response:</p> <p>Success (200) :</p> <pre>{ }</pre>

	<pre>     "message": "Schedule generated successfully",     "schedule": [         {             "task_id": "&lt;UUID&gt;",             "task_name": "Task A",             "start_time": "2025-01-05T08:00:00Z",             "end_time": "2025-01-05T10:00:00Z"         }     ] } Error (400/500): {     "error": "Error message" } </pre>
<b>Implementasi Service</b>	<pre> // Endpoint untuk mengintegrasikan API eksternal app.post('/external/schedule', verifyToken, async (req, res) =&gt; {     try {         const { tasks } = req.body;         // Validasi input tasks         if (!tasks    tasks.length === 0) {             return res.status(400).json({                 status: 'error',                 message: 'Tasks are required',             });         }         // Memastikan setiap task memiliki task_name dan deadline         const invalidTasks = tasks.filter(             (task) =&gt; !task.task_name    !task.deadline         );         if (invalidTasks.length &gt; 0) {             return res.status(400).json({                 status: 'error',                 message: 'One or more tasks have invalid or missing fields: task_name and deadline are required.',             });         }         // Mengirim request ke API eksternal         (https://api.taskly.web.id)         const response = await axios.post('https://api.taskly.web.id/external', {             tasks,         });         // Menangani respon dari API eksternal         const { scheduledTasks, unscheduledTasks } = response.data;         res.status(200).json({             status: 'success',             message: 'Schedule generated successfully',             data: { </pre>

```
        scheduledTasks,
        unscheduledTasks,
    },
},
} );
} catch (error) {
    console.error('Error in external scheduling:', error);
    // Menangani error dari API eksternal atau sistem
    if (error.response) {
        // Error dari API eksternal
        return res.status(error.response.status).json({
            status: 'error',
            message: error.response.data.message || 'Error from external API',
        });
    }
    res.status(500).json({
        status: 'error',
        message: 'Internal Server Error',
    });
}
});
```

## H. Kesimpulan

Website **Piano Progression** digunakan untuk mendukung menyiapkan diri dalam mengikuti ujian piano yang terstruktur, interaktif, dan progres sesuai dengan standar **ABRSM (Associated Board of the Royal Schools of Music)**. *Piano Progression* menyiapkan schedule study plan, progress. Business process yang mencakup perencanaan, pelacakan, umpan balik, dan evaluasi, pengguna dapat menikmati proses belajar yang efektif dan terstruktur. Tech stack yang saya gunakan **React.js**, **Express.js**, **MongoDB** dan dilakukan deployment pada **Vercel** untuk frontend dan **VPS** untuk backend. Fitur-fitur utama, termasuk User Authentication, Landing Page, ABRSM Curriculum, Practice Pathway, Progress Visionary, Flashcard, Feedback and Evaluation. Piano progression juga mengintegrasikan dengan layanan eksternal yaitu **Taskly** yang membantu penjadwalan task.