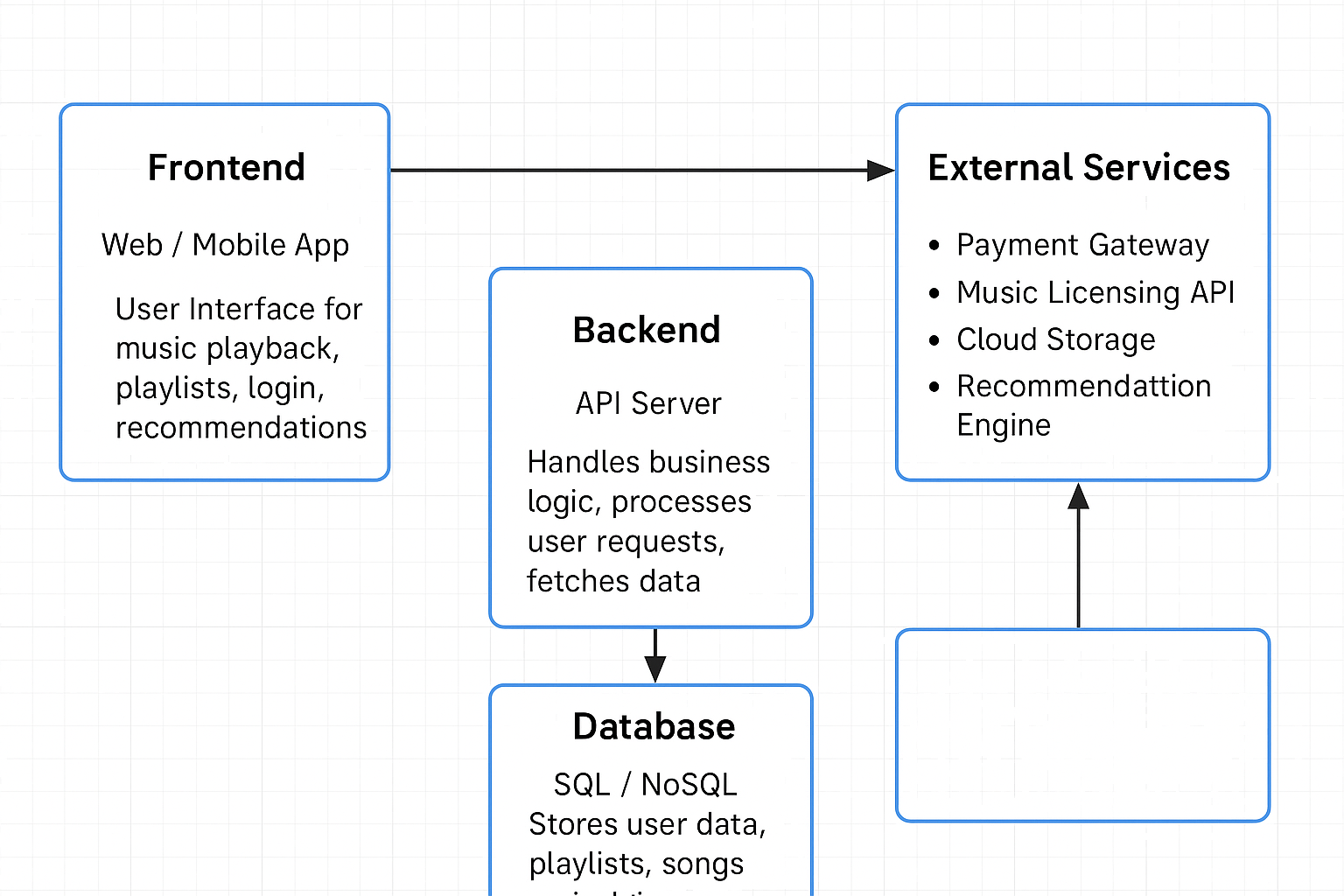
**Project Design Phase-II**

**Technology Stack (Architecture & Stack)**

|  |  |
| --- | --- |
| Date | 15 Apr. 25 |
| Team ID | SWTID1743610409 |
| Project Name | Tune Trails |
| Maximum Marks | 4 Marks |

**Technical Architecture:**

The architecture for **Tune Trail** follows a scalable, microservices-based design to ensure high performance, reliability, and seamless music streaming. Below is the architectural diagram and the technology stack details.



**Table-1 : Components & Technologies:**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Component** | **Description** | **Technology** |
|  | User Interface | Web and mobile interfaces for music streaming | React.js (Web) |
|  | Application Logic-1 | User authentication & profile management | Node.js (Express.js) |
|  | Application Logic-2 | Music streaming & playlist management | Node.js |
|  | Application Logic-3 | Payment processing & subscription logic | Stripe API, PayPal API |
|  | Database | User data, playlists, and metadata storage | PostgreSQL (Relational), Firebase (NoSQL) |
|  | Cloud Database | Scalable cloud-based data storage | AWS DynamoDB / MongoDB Atlas |
|  | File Storage | Audio file storage (MP3, WAV, etc.) | AWS S3 / Google Cloud Storage |
|  | External API-1 | Music metadata (artist, album, genre) | Spotify API / Last.fm API |
|  | External API-2 | Social media integration (sharing features) | Facebook API, Twitter API |
|  | Recommendation Engine | Personalized music recommendations | Python (TensorFlow / Scikit-learn) |
|  | Infrastructure | Deployment & orchestration | AWS EC2, Kubernetes (EKS), Docker |

**Table-2: Application Characteristics:**

| **S.No** | **Characteristics** | **Description** | **Technology** |
| --- | --- | --- | --- |
|  | Open-Source Frameworks | Frameworks used for development | React, Express.js |
|  | Security Implementations | Data encryption, authentication, and OWASP compliance | OAuth 2.0, HTTPS, Firebase Auth |
|  | Scalable Architecture | Microservices with load balancing | AWS |
|  | Availability | Multi-region deployment for redundancy | AWS |
|  | Performance | Low-latency streaming & caching | AWS |

**References:**

<https://c4model.com/>

<https://aws.amazon.com/architecture/>

<https://developer.ibm.com/patterns/>

<https://reactnative.dev/>

<https://www.djangoproject.com/>