

## Project Design Phase-II

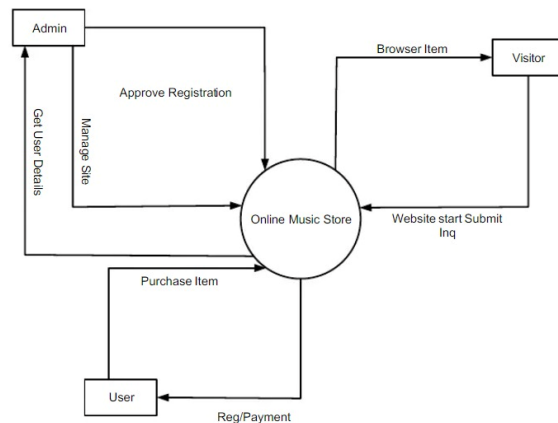
### Technical Architecture (Architecture & Stack)

|              |   |
|--------------|---|
| Date         | 03 Nov 2023                             |
| Team ID      | NM2023TMID04574                         |
| Project Name | How to Create a Reel Design Using Canva |

#### Technical Architecture:

The Deliverable shall include the architectural diagram as below and the information as per the table1 & table 2

#### Example: Order processing during pandemics for offline mode



#### Guidelines:

**Programming Languages:** Use HTML, CSS, and JavaScript as the core languages for the front-end.

**Web Framework:** Consider using a modern web framework like React, Angular, or Vue.js for building the user interface. These frameworks are well-suited for creating interactive and responsive web applications.

**Mobile App Development:** If you plan to offer a mobile app, you can use technologies like React Native or Flutter for cross-platform development. Alternatively, you can develop native apps for iOS and Android using Swift (for iOS) and Kotlin (for Android).

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

| S.No | Component | Description                               | Technology            |
|------|-----------|---|-----------------------|
| 1.   | Front-End | User interface and client-side components | HTML, CSS, JavaScript |

|     |                           |  |  |
|-----|---------------------------|--|--|
| 2.  | Back-End                  | Server-side application, business logic, and APIs        | Node.js, Express.js (JavaScript)<br>- Python (Django or Flask)   |
| 3.  | Database                  | Data storage and management                              | PostgreSQL, MySQL<br>- MongoDB (for unstructured data)   |
| 4.  | Authentication & Security | User authentication, access control, and data security   | OAuth 2.0, OpenID Connect<br>- Digital Rights Management (DRM) for content                                 |
| 5.  | Content Delivery Network  | Efficient content delivery to users                      | CDN services (e.g., Amazon CloudFront, Akamai)   |
| 6.  | Streaming & Playback      | Audio and media streaming                                | HTTP Live Streaming (HLS), DASH<br>- FFmpeg for media processing   |
| 7.  | Recommendation Engine     | Personalized music recommendations                       | Machine learning libraries (e.g., TensorFlow, scikit-learn)  |
| 8.  | Mobile App Development    | Development of mobile apps for iOS and Android           | React Native or Flutter<br>- Swift (iOS) and Kotlin (Android)  |
| 9.  | DevOps & Deployment       | Continuous integration, deployment, and containerization | Jenkins, Travis CI, GitLab CI<br>- Docker for containerization<br>- Kubernetes for container orchestration |
| 10. | Monitoring & Analytics    | Real-time monitoring and user behavior analysis          | Prometheus, Grafana, New Relic<br>- Google Analytics, Mixpanel   |
| 11. | Infrastructure            | Server infrastructure and cloud services                 | AWS, Google Cloud, Azure<br>- Serverless computing for specific tasks                                      |

**Table-2: Application Characteristics:**

| S.No | Characteristics          | Description  | Technology   |
|------|--------------------------|--|--|
| 1.   | Open-Source Frameworks   | List the open-source frameworks used                                       | React, Node.js, Express.js, Django<br>Flask, Ruby, Spring Boot<br>PostgreSQL |
| 2.   | Security Implementations | List all the security / access controls implemented, use of firewalls etc. | Secure APIs, Data Encryption   |
| 3.   | Scalable Architecture    | Justify the scalability of architecture (3 – tier, Micro-services)         | Web Tier, Application Tier, Data Tier  |

| S.No | Characteristics | Description   | Technology  |
|------|-----------------|---|---|
| 4.   | Availability    | Justify the availability of application (e.g. use of load balancers, distributed servers etc.)                            | Compliance and Service Level Agreements (SLAs)    |
| 5.   | Performance     | Design consideration for the performance of the application (number of requests per sec, use of Cache, use of CDN's) etc. | Content Delivery Network (CDN), API Rate Limiting |