**Requirement Gathering and Analysis Phase**

**Technology Stack (Architecture & Stack)**

| Date |  |
| --- | --- |
| Team ID |  |
| Project Name | StreamSavvy |
| Maximum Marks |  |
|  |  |

**Created by – Tejas ravindra koli**

**Architectural Diagram**

(Include an architectural diagram illustrating the flow of data in the movie platform, focusing on the frontend structure and external API usage for fetching movie data.)

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

| **S.No** | **Component** | **Description** | **Technology** |
| --- | --- | --- | --- |
|  | User Interface | Frontend for users to browse, search, and view movies | HTML, CSS, React JS |
|  | Application Logic-1 | Handles movie data fetching and user interactions | JavaScript, React |
|  | External API-1 | | Retrieves movie data and details | | --- |  |  | | --- | | | TMDB API | | --- |  |  | | --- | |

**Table-2: Application Characteristics:**

| **S.No** | **Characteristics** | **Description** | **Technology** |
| --- | --- | --- | --- |
|  | Open-Source Frameworks | React, a JavaScript library for building user interfaces | | React | | --- |  |  | | --- | |
|  | Security Implementations | | Use of HTTPS and secure API calls | | --- |  |  | | --- | | HTTPS |
|  | Scalable Architecture | Scalable React frontend capable of handling user requests | React |
|  | Availability | Cloud deployment ensures high availability and reliability | | Firebase Hosting, AWS | | --- |  |  | | --- | |