Architecture Description

Frontend (Angular)

Components:

- Homepage: Displays anime categories, featured anime, and search options.
- Search & Filter: Allows users to search anime by title, genre, or other criteria.
- Anime Details Page: Displays detailed information about a specific anime, including description, episodes, and streaming options.
- Video Player: Integrated media player for streaming anime content.
- User Profile & Watchlist: Manages user accounts and stores favorite anime shows.

Services:

- AnimeService: Communicates with the backend to fetch anime data via HTTP requests (via Axios or Angular's HttpClient).
- AuthenticationService: Handles user registration, login, and session management (if applicable).

Routing:

- Angular's Router is used for navigating between pages, such as the homepage, search results, and anime details.

Backend (web scraping & server)

Web Scraping Module:

- Scraper (Python/Node.js): A web scraper built using libraries like BeautifulSoup (Python) or Cheerio (Node.js) to fetch anime data from external sites like 9anime or MyAnimeList. The scraper extracts details such as titles, descriptions, image URLs, episodes, and streaming links.
- Scheduler (Optional): A cron job or scheduler to periodically run the scraper and update the anime database with the latest content.

API Layer:

- Flask/Django (Python) or Express (Node.js): Backend server that exposes APIs to serve the scraped anime data to the frontend. These APIs handle requests for searching, viewing anime details, and streaming episodes.
- Database (Optional): A database (e.g., MySQL, MongoDB) can be used to store the scraped anime data for quicker access and to reduce the need for scraping each time.

Integration flow

1. Frontend Angular app sends HTTP requests to the backend API to fetch anime data.
2. The backend web scraping service scrapes data from external websites and stores it in a database or serves it directly.
3. The backend API returns the scraped data to the frontend.
4. Angular components display the anime content, manage search functionality, and provide video streaming.
5. The video player streams anime episodes, either by fetching from external video links or directly hosted files.

4. User Interaction & Personalization

- Users can search for anime, view details, and stream content.
- Authentication and Profile Management can allow users to create accounts, save their watchlists, and rate anime.