

Frontend Developer (F)

General Requirements

- All tasks must be completed using **Nuxt 3** with **Vue 3 Composition API**.
 - Use **TailwindCSS** for all styling.
 - The project must be pushed to a public or private **Git repository** (e.g., GitHub).
 - Use **atomic commits** with meaningful messages to show the step-by-step progress.
 - Provide a **README.md** with setup instructions.
 - Use the provided API at <https://ylnk.site/test/> to fetch all necessary data.
-

Task 1: Player Page UI Implementation

Objective:

Recreate the player page UI based on a provided design (Figma Link). You can choose the mobile or desktop version and only do one.

<https://www.figma.com/design/HHMVHL38ANLXoMRZ5N6hQL/Film-Tarsnak-Test?node-id=6-981&p=f&t=jKiMKc226I50dGxC-0>

Requirements:

- Layout should include:
 - **Movie information section:** title (English and Persian), release year
 - **Rating section:** IMDB, Rotten Tomatoes, Metacritic (if available)
 - **User rating system:** average rating + interactive star voting system
 - **Video player area** (a static image with play button is sufficient for this task)
 - **Episode list sidebar** with episode posters and titles
 - Must be responsive (basic desktop and mobile layout)
 - Load all movie/episode data from the remote API:
 - Endpoint: <https://ylnk.site/test/?action=info&id=2501>
 - Sample response:

```
{ "id": 2501, "title_fa": "سریال کائنات", "title_en": "Kaa'inat", "year": 2024, ... }
```
-

Task 2: Dynamic User Rating Component

Objective:

Build an interactive star-based rating component from scratch.

Requirements:

- The user can rate from 1 to 5 stars.
 - After voting, the user cannot rate again (store lock in localStorage).
 - The component should update and display the new average rating.
 - Use SVG icons for stars, no third-party UI libraries.
 - The component should support customization via props (e.g., maxStars, initialRating).
 - Rating submission should POST to:
 - <https://ylnk.site/test/?action=rate&id=2501>
 - Body: { "rate": 4 }
-

Task 3: Video Player with Subtitle

Objective:

Implement a working video player with HLS playback and a subtitle option.

Requirements:

- Use native <video> with HLS.js or any modern video library (e.g., Plyr).
 - Load video from .m3u8 URL.
 - Load and display subtitle (.vtt) with an option to toggle it on/off.
 - Display current time / duration / playback controls.
 - UI should resemble the player design from Task 1.
 - Video info and subtitle path should be fetched from:
 - Endpoint: <https://ylnk.site/test/?action=stream&id=2501>
 - Sample response: { "video_url": "...m3u8", "subtitle": "subs_kaainat_ep1.vtt" }
-

Final Notes

- You can use any assets or placeholder images.
- The API is publicly available and supports local testing via fetch().
- Ensure all data is loaded dynamically using API calls.
- Your commits should reflect each feature or fix, ideally following Conventional Commits.