

Case Study 2: Modules

Scenario: Movie Streaming Platform – Feature-Based Module Loading

A **movie streaming platform** wants to optimize load times by splitting its functionality into **modules**:

- **User Authentication Module**
- **Movie Catalog Module**
- **Watchlist Module**
- **Recommendation Engine Module** (loaded only when user opens "Recommended")
- **Payment Module**

Project Structure

```
/movie-platform
  auth/
    auth.js
  catalog/
    movies.js
  watchlist/
    watchlist.js
  recommendations/
    recommend.js
  utils/
    logger.js
  app.js
```

Workflow & Description

1. Step 1 – Organizing Code into Modules

- Each major feature (authentication, catalog, watchlist, payment) is implemented as a **separate module**.
- Common utility functions (like logging, formatting, or API requests) are placed in a **utils** module.

2. Step 2 – Using Import/Export Syntax (ES Modules)

- Modules expose only necessary functions/classes via **named exports** or **default exports**.
- **app.js** imports only what it needs to initialize the platform and render the homepage.

3. Step 3 – Dynamic Imports

- Recommendation engine is heavy and rarely used initially.
- It is **dynamically imported** only when the user clicks the "Recommended" section.
- Reduces initial load time and improves performance.

4. Step 4 – Lazy Loading Additional Features

- Other optional modules (e.g., Payment gateway for subscription) are also loaded dynamically when triggered.

5. Benefits

- Cleaner, modular code structure → easy maintenance and testing.
- Faster initial page load → only critical modules loaded upfront.
- Dynamic imports allow feature-on-demand loading.

Key Concepts Demonstrated

- Named and default exports
- Modular code organization for scalability
- Dynamic imports for performance optimization
- Dependency isolation between modules

Sample Output (User Interaction Flow)

```
INFO: App initialized successfully
INFO: User logged in: ID 202
INFO: Movie catalog loaded: ["Inception", "The Matrix",
  "Interstellar"]
INFO: Watchlist loaded: ["Inception", "Interstellar"]
INFO: User clicked 'Recommended' → loading recommendation
  engine dynamically
INFO: Recommended movies: ["Tenet", "Interstellar", "Dune"]
INFO: User added "Dune" to watchlist
INFO: Payment module loaded for subscription checkout
INFO: Payment successful: Transaction ID 78901
```

Note:

Modules like **Recommendation Engine** and **Payment Module** were **not loaded initially**, demonstrating dynamic imports.