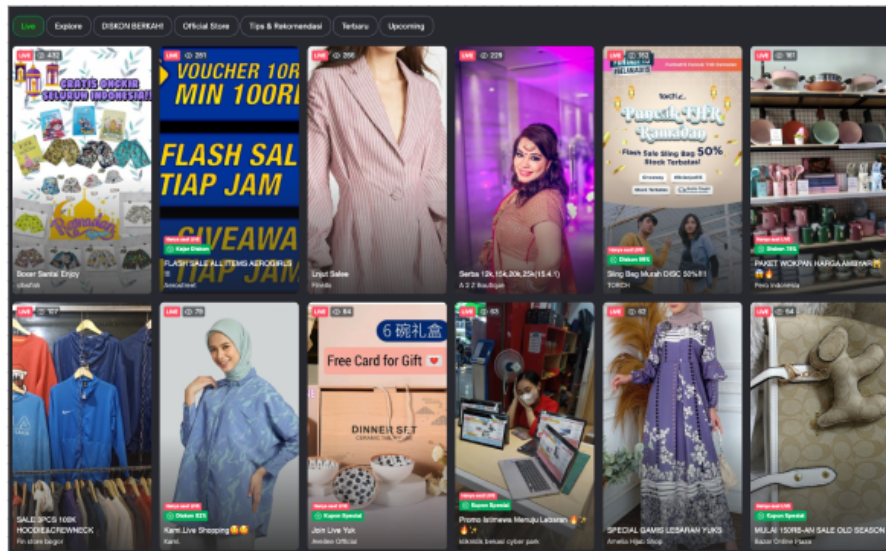


Fullstack Track Final Project

Wireframe

Video List



Video Detail



Instruction

1. You are build a [tokopedia play](#) clone, we give you guideline on wireframe section. You can modify wireframe as long you fulfill tech requirement and features
2. You can use any UI library of your choice. Aesthetics is subjective but imagine you are the user of your app and you want the best.
3. Pick at least 5 video to sell product in YouTube
4. Features
 - a. User can open, at least two page
 - i. Home
 - ii. Video Detail Page
 - b. User can see video list with thumbnail from YouTube in home page
 - c. User can click each video and go to video detail page
 - d. User can see list products, embed YouTube, list comment, and form submit comment
 - e. User just need input name and comment when submit comment in form
 - f. User can see their comment in list comment section after success submit comment
5. Technical Requirements
 - a. API: Follow API Minimum section and restful standard
 - b. Database: Create DB to support list API needed
 - c. React Hooks: create custom hooks at least two
 - d. Use React Router as router navigation between page
6. Add a readme which includes these following things (at minimum). Take a look at <https://www.makeareadme.com/> as the references.
 - a. Project name and description
 - b. Features (list down also additional features if any)
 - c. How to install & run
 - d. Schema Database

Deliverables:

Mid Term Project (Backend Only)

1. You need to submit these things for your assignment
 - a. The Github public repo URL.
 - b. Readme on Repo, include
 - i. Explain your database structure
 - ii. Explain API structure

- iii. List API request and response (preference format [gist](#))
- iv. How to run in local (**make sure grader can run your API in grader local machine after following your “how to run”!**)

Final Project (Backend and Frontend)

1. You need to submit these things for your assignment
 - a. The hosted URL of your project. (Backend and Frontend)
 - b. The Github repo URL.
 - c. Compressed (.zip format) of your project.
2. You can't update your code (in production or github) after you submit it

API Minimum

You require to create this API to fulfill our requirement, but you can add if needed

Title	Response	Argument/Payload	Type
Video Thumbnail List	<ul style="list-style-type: none"> - VideoID - Url Image Thumbnail 		GET ▾
Product List	<ul style="list-style-type: none"> - ProductID - Link Product - Title - Price 	<ul style="list-style-type: none"> - VideoID 	GET ▾
Comment List	<ul style="list-style-type: none"> - Username - Comment - timestamp 	<ul style="list-style-type: none"> - VideoID 	GET ▾
Submit Comment	<ul style="list-style-type: none"> - Success - Fail 	<ul style="list-style-type: none"> - Username - Comment - VideoID 	POST ▾

Bonus:

Add one feature outside mentioned from requirement above, example

- add user profile picture and username on the top right corner of the page
- Use websocket or server side event for push real time data for comment list
- Add feature search

Scoring Components:

1. App is deployed into production, and anyone with an internet connection can access it.
2. All features are working.
3. All technical requirements are met.
4. Concepts
 - a. Clean code: folder structure, naming (files, variable, component, functions), all should be consistent
 - b. Component structure: your ability to abstract problems into components. It should be sensible (not too nested as in having 4 levels of hierarchy), extensible, and maintainable
 - c. Database Structure: your ability to create database with efficient schema
 - d. API Best Practices