1 Using real time protocol like websocket(socket.io) (like smart cable) to create channel and transfer api
Data or in another way you can use GraphQL to optimize API requests to query large
Only needed multi value /api data in minimize request
But you can optimize api also like minimize request payload ,correct cache using such as Radis , pagination cut off,and batch api into group also use API as needed only (lazy load,debounce)

2 Create full test and develop strategy for frontend (UI flows) backend (postman testing)that
Support business case flow (like purchasing and subscribe membership/Social SSO) support
Various case i.e. big data , edge case and monitor like grafana

3 use callback
For callback function ; to avoid re call function of parent or whole parent component (cause infinity loop) when render component again and avoid
redundant code (same function) in SPA (must use with use memo)

```
test('renders name', async () => {
  axios.get.mockResolvedValue({ data:{ user:{name: 'warat.narat@gmail.com',email: 'warat.narat@gmail.com'  }
});

  render(<UserProfile userId="1" />);

  const userNameElement = await screen.findByText('Warat');
  expect(userNameElement).toBeInTheDocument();
});

test('renders  email', async () => {
  axios.get.mockResolvedValue({ data: { user:{name: 'warat.narat@gmail.com',email: 'warat.narat@gmail.com'
}} });

  render(<UserProfile userId="1" />);

  const userNameElement = await screen.findByText(‘Email: warat.narat@gmail.com');
  expect(userNameElement).toBeInTheDocument();
});
```