* Create me a React project and do the following:

1. Create a latest version of react project
2. The project must have dropdown box and a input field.
3. The dropdown component will have items like twitter, facebook, instagram, discord, twitter.
4. Select an item from the dropdown and after selecting the corresponding url has to be given in the field.
5. Eg: If the user selects facebook option , then facebook has to be given in the input field.
6. If the user gives wrong url Eg: If the user selects facebook option and gives telegram url in the input field then a error message must be shown below the input field.
7. Create this project as reusable components
8. Load the dropdown items and regex using for url validation from the file name called contant.ts

* Analyze this project and generate tests for every possible edge

case.

* Think like a hacker trying to break the system.Consider:

1. Null/undefined/empty inputs
2. Extremely large or small numbers
3. Special characters and Unicode
4. Concurrent access scenarios
5. Memory limitations
6. Network timeouts
7. Invalid data types
8. Boundary conditions
9. Race conditions
10. Security vulnerabilities

1. Input Validation Edge Cases:

// Empty Input Cases

- Empty platform selection

- Empty URL input

- Invalid URL format (malformed URLs)

// URL Format Edge Cases

- Invalid URL format (non-URL strings)

- URLs with special characters

- URLs with Unicode characters

- Extremely long URLs (over 2048 characters)

2.Platform-Specific Edge Cases:

// Platform Validation

- Testing all supported platforms (Facebook, Twitter, Instagram, Discord)

- Platform-specific URL patterns

- Case sensitivity in platform names

- Platform switching behavior

3.State Management Edge Cases:

// Error State Management

- Error message clearing on platform change

- Error message clearing on URL change

- Multiple error states (platform + URL errors)

// Success State Management

- Success message clearing on platform change

- Success message clearing on URL change

- Success state with alert confirmation

4.Security Edge Cases:

// URL Security

- XSS attempt prevention

- SQL injection attempt prevention

- Special character handling

- Unicode character handling

5. Component Interaction Edge Cases:

// Callback Handling

- onChange callback with valid input

- onChange callback with invalid input

- Callback parameter validation

// Event Handling

- Rapid input changes

- Rapid platform changes

- Form submission without required fields

6. Browser Environment Edge Cases:

// Alert Handling

- Alert in browser environment

- Alert in test environment

- Alert message content verification

7. UI State Edge Cases:

// Message Display

- Error message visibility

- Success message visibility

- Message content accuracy

- Message clearing behavior

// Form State

- Disabled state handling

- Loading state handling

- Form reset behavior

8. Performance Edge Cases:

// Input Performance

- Rapid input changes

- Rapid platform changes

- Long URL handling

- Multiple validation cycles

9. Integration Edge Cases:

// Component Integration

- Parent-child component communication

- Props validation

- Event propagation

- State synchronization

10. Error Recovery Edge Cases:

// Error Recovery

- Recovery from invalid input

- Recovery from network errors

- Recovery from validation errors

- State reset behavior