

# Improvement Report

## Test Bank

### Components Developed:

1. Case Study Simulation
  - Created interactive case study scenarios
  - Developed simulation interface for user interaction
  - Built decision-making pathways for different user choices
2. Interactive Exercises
  - Designed hands-on learning activities
  - Created step-by-step problem-solving modules
  - Implemented immediate feedback mechanisms
3. Knowledge Tests
  - Developed assessment modules with scoring system
  - Created multiple question formats (multiple choice, true/false, matching)
  - Built progress tracking and results display

New design:



Main page: <https://lexue.bit.edu.cn/course/view.php?id=16945&section=34>

<https://lexue.bit.edu.cn/mod/quiz/view.php?id=514878>

<https://lexue.bit.edu.cn/mod/url/view.php?id=513390>

<https://lexue.bit.edu.cn/mod/url/view.php?id=513391>

<https://lexue.bit.edu.cn/mod/url/view.php?id=513398>

<https://lexue.bit.edu.cn/mod/url/view.php?id=513399>

<https://lexue.bit.edu.cn/mod/url/view.php?id=513400>


<https://lexue.bit.edu.cn/mod/url/view.php?id=513392>

<https://lexue.bit.edu.cn/mod/url/view.php?id=513403>

<https://lexue.bit.edu.cn/mod/hvp/view.php?id=515209>

<https://lexue.bit.edu.cn/mod/hvp/view.php?id=513365>

Case Study Simulation [Website/haven't published]

 **SDLC Stage Sorter**

Drag and drop the Software Development Life Cycle stages into the correct order.

Unplaced Stages


Planning, Design, Maintenance, Development, Testing, Deployment, Requirements

Correct Order

Drop stages here

Check Order

Reset

 **Recursion Call Stack Visualizer**

See how the call stack works when calculating factorial(3) using recursion.

n = 3


function factorial(n) {  
 if (n === 0) {  
 // Base Case  
 return 1;  
 } else {  
 // Recursive Step  
 return n \* factorial(n - 1);  
 }  
}

Call Stack

call factorial(0)  
call factorial(1)  
call factorial(2)

▶ Calculating...

↺ Reset

 **Hospital Management System**

A simplified database simulation for managing patient records and appointments.

Patient Database

Add New

John Doe (P001)  
Jane Smith (P002)

Add New Patient

Cancel

Patient Name

Appointment Time

mm/dd/yyyy --:-- --

Save Patient

## E-commerce Platform Simulation

Explore how an e-commerce site works, including adding items to a cart and seeing personalized recommendations.

### Products



Stylish Sneakers

\$79.99

Add to Cart



Wireless Headphones

\$129.99

Add to Cart



Modern Backpack

\$59.99

Add to Cart

## Secure Banking App

A simple simulation of a mobile banking app, focusing on secure login and basic transactions.



Username

user123

Password

.....

→ Log In

## Food Delivery System

Simulates the real-time order tracking of a food delivery app, powered by cloud services and APIs.

### Your Order Status



Order Placed



Restaurant is Preparing



Rider is on the way



Delivered

Tracking Order...

## Logic Gate Simulator

Experiment with basic logic gates, the building blocks of digital circuits. Change the inputs and the gate type to see the output.

Input A



Input B



XOR



Output



0

## Simple Expert System

Diagnose a simple ailment based on symptoms using a set of IF-THEN rules. This mimics a basic medical expert system.

### Select Symptoms (Facts)

☒ Persistent Cough

☐ Fever

☒ Fatigue


☒ Headache

Run Diagnosis

Reset

### Diagnosis Result

#### Migraine

 Reasoning:

- System matched the following symptoms: Headache, Fatigue.

🧩 Search Algorithm Visualizer

Visualize how Breadth-First Search (BFS) and Depth-First Search (DFS) explore a maze to find the goal.

Breadth-First Search (BFS) ▾

▶ Run Simulation

↺ Reset

● Start

● End

● Wall

● Visited

● Path

⚖️ Legal Reasoning Simulator

Determine voting eligibility by applying legal statutes as logical IF-THEN rules.

Set the Facts

- ☒ Is a Citizen?
- ☐ Is Registered to Vote?

Age: 17

Conclusion

Ineligible to Vote

Reasoning Steps:

✓ Person IS a citizen.

✗ Person is 17 years old, which is less than 18.

Evaluate

Reset

🛡️ Fraud Detection System

A hybrid system where a software layer processes transactions and a symbolic AI layer applies fraud-detection rules.

Incoming Transactions

Amount: \$50.25	Location: Local	Time: Normal
✔ Transaction appears normal.		
Amount: \$6500.00	Location: Local	Time: Normal
⚠ Flagged: High transaction amount for a local transfer.		
Amount: \$120.00	Location: Local	Time: Normal
✔ Transaction appears normal.		
Amount: \$250.00	Location: International	Time: Normal
✔ Transaction appears normal.		
Amount: \$300.50	Location: International	Time: Odd Hours
⚠ Flagged: Unusual international activity during odd hours.		

Run Fraud Detection Rules