# Task 3: Blockchain-Based Experiment Platform – Ultimatum Game

## **Written Design of the Experiment:**

### Experiment Name: Ultimatum Game on Blockchain

### Objective:

To implement a simple economic experiment where Player A propose a division of 1000 satoshis, and Player B either accepts or rejects the offer, using blockchain smart contracts to automate enforcement.

### Participants:

* Player A: Proposes a split.
* Player B: Accepts or rejects the proposed split.

### Rules:

* If Player B accepts -> The funds are split according to Player A’s proposal.
* If Player B rejects -> Neither player gets anything.

### Blockchain Usage:

* Smart contract enforces the rules.
* No trusted third party is needed – the blockchain guarantees the payout or loss.

## **Smart contract Logic (Simple pseudocode)**

A screenshot of a computer program

AI-generated content may be incorrect.

## **Compilation and Deployment Evidence**

### Compilation Note:

Remix IDE was used to compile and simulate the Ultimatum Game smart contract due to technical issues accessing sCrypt IDE.  
The contract was compiled successfully, and test transactions were performed using local JavaScript VM.

I have inserted my screenshots below as evidence to my smart contract.

1. Remix IDE showing "Contract Compiled Successfully"

A screenshot of a computer program

AI-generated content may be incorrect.

1. proposeSplit(400) executed and proposer/responder shares updated

A screenshot of a computer program

AI-generated content may be incorrect.

1. respond(true) showing “Accepted: Split executed”

A screenshot of a computer program

AI-generated content may be incorrect.

## **Front-End Simulation:**

A simple HTML front-end was created to simulate user interaction with the smart contract.  
 Player A can input a proposed amount, and Player B can accept or reject via buttons.  
 Results are displayed based on the players' actions.

**Front-end screenshots:**

1. Webpage with Player A input field and Accept/Reject buttons:

A screenshot of a computer

AI-generated content may be incorrect.

1. Result after clicking “Accept” (Showing split):

A screenshot of a computer

AI-generated content may be incorrect.

1. Result after clicking "Reject” (Showing zero payout):

A screenshot of a computer

AI-generated content may be incorrect.

## 5. **Reflections and Learnings:**

* This task improved my understanding of how smart contracts automate economic experiments securely.
* I learned to simulate contracts in Remix IDE when access to sCrypt IDE was unavailable.
* Designing a simple front-end allowed me to visualize user interactions with blockchain contracts.
* Blockchain ensures fairness and transparency without needing central authority.