

Output – Experiment 1

The screenshot shows a Windows desktop environment with the Visual Studio Code (VS Code) application open. The title bar of the VS Code window displays "bank----- Added". The left sidebar contains a file tree with project files: build, contracts, migrations, node_modules, scripts, test, commands, package-lock.json, package.json, ReentrantCaller.call-graph.dot, and truffle-config.js. The main area shows a terminal window with the following command history:

```
PS C:\Users\maria\bank----- Added> npm run reenter:v1
Caller balance: 3 ETH
● PS C:\Users\maria\bank----- Added> npm run deposit:v2

> bank@1.0.0 deposit:v2
> truffle exec scripts/deposit.js --network development --v2 --amount 2

Using network 'development'.

Deposited 2 ETH into BankV2 from 0x8985D90B2DAeE60eb7a99c7e4f348a220F93563e
Contract balance is now: 2 ETH
● PS C:\Users\maria\bank----- Added> npm run reenter:v2

> bank@1.0.0 reenter:v2
> truffle exec scripts/executeReentry.js --network development --v2 --deposit 1

Using network 'development'.

ReentrantCaller deployed at 0xA498392f43dADBfc3e14c641fae308D1622b7544, targeting BankV2
execute() reverted (expected for BankV2). Message: Transaction: 0x2704020dfef59c36dd65673981c663ff9d4f0ca732c80e257d
0f621af298b16f exited with an error (status 0). Reason given: Transfer failed.
    Please check that the transaction:
        - satisfies all conditions set by Solidity `require` statements.
        - does not trigger a Solidity `revert` statement.

Bank balance: 2 ETH
Caller balance: 0 ETH
PS C:\Users\maria\bank----- Added>
```

The bottom status bar shows various system icons and the time "19:38".

Output – Experiment 2

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** On the left, it shows a project structure for "EXPLORER: ACCESS_CON..." with files like build, contracts, migrations, node_modules, scripts, test, package-lock.json, package.json, and truffle-config.js.
- Terminal:** The main area displays the output of npm commands run mint, revokeRole, burn, and grantRole against a Truffle network named "development".

```
PS C:\Users\maria\access_control ----- Added> npm run mint
> access_control@1.0.0 mint
Using network 'development'.

Minting 100 points to 0xEA5fE8f622ac41EE29dc7c79169F1a4b1DC1941C as 0x8985D90B2DAeE60eb7a99c7e4f348a220F93563e...
Tx: 0xe7b2b8b201a8ad01130a3d58130a5c52fb01f85df62ef6dcbb977d1880bf99e8
points[0xEA5fE8f622ac41EE29dc7c79169F1a4b1DC1941C] = 100
PS C:\Users\maria\access_control ----- Added> npm run revokeRole

> access_control@1.0.0 revokeRole
> truffle exec scripts/revokeRole.js

Using network 'development'.

[dotenv@17.2.1] injecting env () from .env -- tip: ⚡ observe env with Radar: https://dotenvx.com/radar
Revoking BURNER_ROLE from 0x1B92Cf1c02F66C66dfd99CF3F29A926b9839a702...
Tx: 0x945f94ae43eae049e34c5b3d26eaeebf4b0a7a434566bceec586879cf54cc55f
BURNER_ROLE revoked from 0x1B92Cf1c02F66C66dfd99CF3F29A926b9839a702
PS C:\Users\maria\access_control ----- Added> npm run burn

> access_control@1.0.0 burn
> truffle exec scripts/burn.js

Using network 'development'.

0x1B92Cf1c02F66C66dfd99CF3F29A926b9839a702 does NOT have BURNER_ROLE. Burn blocked (as expected).
Grant it back with:
ROLE_NAME=BURNER_ROLE TARGET=0x1B92Cf1c02F66C66dfd99CF3F29A926b9839a702 npm run grantRole
PS C:\Users\maria\access_control ----- Added>
```
- Status Bar:** At the bottom, it shows the status bar with "Ln 1, Col 1", "Spaces: 2", "UTF-8", "LF", "JSON", "Go Live", "Prettier", and system icons for battery, signal, and date/time (23-10-2025, 19:46).

Output – Experiment 3

The screenshot shows a Microsoft Visual Studio Code (VS Code) interface with the following details:

- File Explorer (Left):** Shows the project structure with files like `build`, `contracts`, `migrations`, `node_modules`, `scripts` (containing `checkBalance.js`, `eventListener.js`, `mint.js`, `queryEvents.js`, `transfer.js`), `test` (containing `.env`), and configuration files like `package-lock.json`, `package.json`, `query`, `README.md`, and `truffle-config.js`.
- Search Bar (Top):** Contains the query "analytics ----- Added".
- Code Editor (Center):** Displays the `package.json` file with the following content:

```
8  },
9  "scripts": {
10   "deploy": "truffle migrate --reset",
11   "mint": "truffle exec scripts/mint.js",
12   "transfer": "truffle exec scripts/transferTokens.js",
13   "listen": "truffle exec scripts/eventListener.js",
14   "test": "truffle test"
15 }
```
- Terminal (Bottom):** Shows the command `npm run listen` being executed, followed by the output of the Truffle application running on the `development` network. The output includes connecting to MongoDB, listening on port 5777, starting event listeners, and successfully saving a mint event to MongoDB with the transaction hash `0xe71b0a3674ee174e578ead8fe15ae357288e2f3a1adcb51ba1d718ee7914410e`.
- Bottom Status Bar:** Includes icons for file status (4 changes), search, connect, line 11, column 39, spaces 2, UTF-8, LF, JSON, Go Live, Prettier, and date/time (24-10-2025, 00:09).

Output – Experiment 4

The screenshot shows a Microsoft Visual Studio Code (VS Code) interface with the following details:

- File Explorer:** On the left, it displays a file tree for a project named "Event-trigger". The tree includes folders for artifacts, cache, contracts (containing Lock.sol and ProductManager.sol), ignition, node_modules, scripts (containing createProduct.js, eventListener.js, getBalance.js, getProductDetails.js, markedDelivered.js, purchaseProduct.js, withdraw.js), test (containing EventTrigger.js, Lock.js, .env, .gitignore, hardhat.config.js), and files package-lock.json, package.json, and README.md.
- Code Editor:** The main editor area shows a file named "package.json" with the following content:

```
5   "scripts": {  
6     --network localhost",  
7     "create-product": "hardhat run scripts/createProduct.js",  
8     "purchase-product": "hardhat run scripts/purchaseProduct.js",  
9     "mark-delivered": "hardhat run scripts/markDelivered.js",  
10    "withdraw": "hardhat run scripts/withdraw.js",  
11    "get-product": "hardhat run scripts/getProductDetails.js"  
12  },  
13  "Keywords": []
```
- Terminal:** The bottom terminal window shows the following command-line interaction:

```
PS C:\Users\maria\Event-trigger> hardhat run .scripts\eventListener.js  
hardhat : The term 'hardhat' is not recognized as the name of a cmdlet, function, script file,  
or operable program. Check the spelling of the name, or if a path was included, verify that the  
path is correct and try again.  
At line:1 char:1  
+ hardhat run .scripts\eventListener.js  
+ ~~~~~~  
+ CategoryInfo          : ObjectNotFound: (hardhat:String) [], CommandNotFoundException  
+ FullyQualifiedErrorId : CommandNotFoundException
```



```
PS C:\Users\maria\Event-trigger> npx hardhat run .scripts\eventListener.js  
[dotenv@0.17.2.1] injecting env (4) from .env -- tip: 🛡 encrypt with Dotenvx: https://dotenvx.com  
Listening for blockchain events... Press Ctrl+C to stop.
```

[PRODUCT CREATED]
Product ID: 2
Name: kitbag
Price: 0.5 ETH
Block: 4
TX: 0xe5176fb67ef72ae31bec7bf402f82d03d5533579115e6480eacf0a3b9aec2ef1
- Right Panel:** A "Build with agent mode" panel is open, showing AI-related options like "Generate Agent Instructions".
- Bottom Bar:** The taskbar at the bottom shows various pinned icons for tools like GitHub, Microsoft Edge, Google Chrome, and others.

Output – Experiment 5

The screenshot shows the Postman application interface. The top navigation bar includes Home, Workspaces, Explore, a search bar, and account options (Sign In, Create Account). A message at the top states: "You are using the Lightweight API Client, sign in or create an account to work with collections, environments and unlock all free features in Postman."

The main area displays a "History" section on the left with a list of recent requests, including several GET and POST requests to localhost:3000. A specific POST request to "http://localhost:3000/mint" is selected in the center.

The "Body" tab is active, showing the JSON payload sent to the endpoint:

```
1 {  
2   "to": "0x70997970C51812dc3A010C7d01b50e0d17dc79C8",  
3   "amount": "250"  
4 }  
5  
6 }
```

The response status is 200 OK, with a response time of 508 ms and a response size of 422 B. The response body is also displayed:

```
1 {  
2   "success": true,  
3   "message": "Minted 250 tokens to 0x70997970C51812dc3A010C7d01b50e0d17dc79C8",  
4   "txHash": "0x3df228f6861470abc8f461bed0967547679ba7f1273dacea3f2f07f7e5fd4cab",  
5   "blockNumber": 1  
6 }
```

The bottom of the screen shows the Windows taskbar with various pinned icons and the date/time: 23-10-2025 23:28.

Output – Experiment 6

The screenshot shows a Microsoft Visual Studio Code (VS Code) interface with the following details:

- File Explorer (Left):** Shows the project structure under "EXPLORER: IPFS".
 - client:** .next, app, components, hooks, lib, node_modules, public, styles, .gitignore, components.json, next-env.d.ts, next.config.mjs, package-lock.json, package.json (selected), pnpm-lock.yaml, postcss.config.mjs, tsconfig.json.
 - server:** config, controllers, database, models, node_modules, routes, services, .gitignore, package-lock.json, package.json, server.js, test-flow.js, test.txt.
- Code Editor (Top Center):** Displays the contents of "package.json" in the "client" folder.

```
1 {  
2   "name": "my-v0-project",  
3   "version": "0.1.0",  
4   "private": true,
```
- Terminal (Bottom Left):** Shows the command "node server.js" being run and its output:

```
PS C:\Users\maria\Downloads\Ipfs\server> node server.js  
✓ Documents table is ready.  
PS C:\Users\maria\Downloads\Ipfs\server> node server.js  
Database path: C:\Users\maria\Downloads\Ipfs\server\database\documents.db  
Server running on http://localhost:3000/  
Test the server at http://localhost:3000/  
Documents API at http://localhost:3000/api/documents/  
✓ Connected to SQLite database.  
✓ Documents table is ready.  
POST /api/documents/upload  
Upload request received  
File: Media.jpeg (190766 bytes)  
Adding to IPFS...  
Uploading file to Pinata IPFS... Media.jpeg  
File uploaded to Pinata successfully!  
IPFS Hash (CID): QmflWv6RkNgZSm4X5eaRA2nEEPFT3wKoY7L5HcyvtqpuP2U  
Pin Size: 190780  
✓ IPFS Result: {  
cid: 'QmflWv6RkNgZSm4X5eaRA2nEEPFT3wKoY7L5HcyvtqpuP2U',  
pinSize: 190780,  
timestamp: '2025-10-23T17:51:35.256Z'  
}  
Saving to database...  
✓ Document saved to database with ID: 10
```
- Bottom Status Bar:** Shows the current file is "main*", has 6 changes, 0 errors, 0 warnings, and is connected. It also displays system icons for battery, signal, and network, along with the date and time (23-10-2025, 23:21).

Output – Experiment 7

The screenshot shows a Microsoft Visual Studio Code (VS Code) interface with the following details:

- File Explorer:** On the left, it shows a project structure for "WEB3-LIBRARY". The "server" folder contains files like "addBooks.js", "truffle-config.js", "package.json", ".gitignore", ".secret", "package-lock.json", and ".env".
- Editor:** The main editor area displays the "package.json" file, specifically the "scripts" section. The code includes various Truffle commands for deployment and testing across different networks (ganache, sepolia).

```
    "scripts": {
      "migrate:reset": "truffle migrate --reset",
      "migrate:ganache": "truffle migrate --network ganache --reset",
      "migrate:sepolia": "truffle migrate --network sepolia --reset",
      "test": "truffle test",
      "test:ganache": "truffle test --network ganache",
      "console": "truffle console",
      "console:ganache": "truffle console --network ganache",
      "console:sepolia": "truffle console --network sepolia",
      "add-books": "node scripts/addBooks.js",
      "add-books:local": "NETWORK=development node scripts/addBooks.js",
      "add-books:ganache": "NETWORK=ganache node scripts/addBooks.js",
      "add-books:sepolia": "NETWORK=sepolia node scripts/addBooks.js",
      "verify:sepolia": "truffle run verify Web3Library --network sepolia",
      "deploy:local": "npm run migrate:reset",
      "deploy:ganache": "npm run migrate:ganache && npm run add-books:ganache",
      "deploy:sepolia": "npm run migrate:sepolia && npm run add-books:sepolia"
    }
```
- Terminal:** The bottom panel shows the terminal output for two commands:
 - migrate:reset:** Shows deployment statistics: Total deployments: 2, Final cost: 0.005524918714439195 ETH.
 - add-books:** Shows the execution of "node scripts/addBooks.js". It uses a wallet address (0xcf5c32Fdc7d26e07807ac7b5542b889747F1a5EF) and adds books in bulk. An estimated gas for bulk add is 608549, and the bulk transaction hash is 0x10612cae862609793c140e0494f05d9b9535d6cf3e9c7864e05d451150906d6f. A success message indicates the bulk add was successful with gas used: 602599.
- GitHub Copilot Chat:** A sidebar on the right shows a message about reaching the monthly chat quota and an option to "Upgrade to GitHub Copilot Pro".
- Bottom Status Bar:** The status bar shows the user's name (Arunmani21), the date (1 month ago), the current line and column (Ln 9, Col 34), spaces (2), encoding (UTF-8), and the date and time (23-10-2025, 23:37).

Output – Experiment 8

The screenshot shows the Postman interface with the following details:

- Header Bar:** Home, Workspaces, Explore, Search Postman, Sign In, Create Account.
- Message Bar:** You are using the Lightweight API Client, sign in or create an account to work with collections, environments and **unlock all free features in Postman**.
- Left Sidebar (History):** Shows a list of recent API calls, mostly GET requests to `http://localhost:3000/student` and POST requests to `http://localhost:3000/register`, with a note of 8 new items.
- Current Request:** `POST http://localhost:3000/register`
- Request Details:**
 - Method:** POST
 - URL:** `http://localhost:3000/register`
 - Params:** (selected tab)
 - Headers:** (9 items)
 - Body:** (green dot)
 - Pre-request Script:** (green dot)
 - Tests:** (green dot)
 - Settings:** (green dot)
 - Cookies:** (blue link)
- Response Preview:** (JSON)

```
1 "success": true,
2 "txHash": "0x943bb4374ae0a99746a3bfa0514e59ecc3c9ded24e36b4ed39a8f85ae4f8d339"
```
- Bottom Navigation:** Console, Not connected to a Postman account, a Windows taskbar with various icons, and system status indicators.