Staking Lab Handout

Deploy and operate the SimpleStaking contract on Base Sepolia, then drive it from the browser lab at blockchain-camp.lovable.app/lab. Print this sheet and tick each checkpoint as you go.

FIELD	VALUE
Network	Base Sepolia testnet
Chain ID	84532 (0x14a34)
RPC URL	https://sepolia.base.org
Explorer	https://sepolia.basescan.org
Wallet	MetaMask (EVM / secp256k1, EIP-1193 provider)
Contract	SimpleStaking.sol — native ETH staking, linear rewards

Before you start

- Create a **fresh MetaMask account** used only for testnets. Never reuse a mainnet key.
- Fund it from a Base Sepolia faucet — you need gas plus a little ETH to stake and to fund rewards.
- Download **SimpleStaking.sol**, the ABI, and the test files from the lab page.
- Optional: open the SolForge Solidity IDE at solidity-ide-v1.lovable.app to read the contract.

Contract surface

```
function stake() payable                // stake msg.value
function unstake(uint256 amount)        // withdraw principal
function claimRewards()                 // pull accrued rewards
function pendingRewards(address) view returns (uint256)
function totalStaked() view returns (uint256)
function rewardRate() view returns (uint256)  // wei per staked wei per second
```

Rewards accrue linearly: **pending = amount x rewardRate x elapsedSeconds**. Rewards are paid from the contract's own ETH balance, so the owner must send ETH to the contract before any claim can succeed.

Local setup — Hardhat

```
mkdir staking && cd staking && npm init -y
npm install --save-dev hardhat @nomicfoundation/hardhat-toolbox
npx hardhat init            # choose 'Create a JavaScript project'
# copy SimpleStaking.sol -> contracts/  and SimpleStaking.test.js -> test/
npx hardhat compile
npx hardhat test
```

```
# hardhat.config.js
networks: { baseSepolia: { url: 'https://sepolia.base.org',
                           accounts: [process.env.PRIVATE_KEY] } }

# scripts/deploy.js
const s = await ethers.deployContract('SimpleStaking', [1_000_000_000n]);
await s.waitForDeployment(); console.log(await s.getAddress());

npx hardhat run scripts/deploy.js --network baseSepolia
```

Local setup — Foundry

```
curl -L https://foundry.paradigm.xyz | bash && foundryup
forge init staking && cd staking
# copy SimpleStaking.sol -> src/  and SimpleStaking.t.sol -> test/
forge build
forge test -vv

export PRIVATE_KEY=0x...    # test-only key funded with Base Sepolia ETH
forge create src/SimpleStaking.sol:SimpleStaking \
  --rpc-url https://sepolia.base.org \
  --private-key $PRIVATE_KEY \
  --constructor-args 1000000000
```

What the test suite proves

- Staking increases the caller's recorded stake and **totalStaked**.
- Rewards grow with elapsed time and reset after a claim.
- Unstaking more than the recorded stake reverts.
- Claiming with an unfunded reward pool reverts instead of silently paying zero.
- Only the owner can fund or withdraw the reward pool.

Guided checklist

The lab page auto-detects each of these. Mark your own result here as you watch the UI flip to PASS.

01	**Wallet installed & connected** MetaMask injected, eth_requestAccounts returned an address.	PASS / FAIL
02	**Switched to Base Sepolia (chainId 84532)** wallet_switchEthereumChain, or wallet_addEthereumChain if unknown.	PASS / FAIL

03	**Contract deployed & address loaded** Paste the deployed address into the lab; ABI defaults are provided.	PASS / FAIL
04	**Reward pool funded** Contract ETH balance is greater than totalStaked.	PASS / FAIL
05	**Stake tx sent** stake() called with a non-zero msg.value; tx confirmed on Basescan.	PASS / FAIL
06	**Rewards accrued** pendingRewards(you) returns a value above zero after some seconds.	PASS / FAIL
07	**Rewards claimed** claimRewards() succeeded and pending reset to ~zero.	PASS / FAIL
08	**Unstake tx sent** unstake(amount) returned principal to your wallet.	PASS / FAIL

If a step fails

SYMPTOM	LIKELY CAUSE & FIX
Nothing happens on Connect	No injected provider. Install MetaMask, reload, and disable competing wallet extensions.
Wrong network banner stays	Chain not added yet. Approve the wallet_addEthereumChain prompt with RPC https://sepolia.base.org.
Call reverts with no reason	Address points to a different contract or the ABI does not match. Re-check the deployed address on Basescan.
Claim reverts	Reward pool empty. Send ETH to the contract from the owner account, then retry.
Insufficient funds for gas	Top up from a Base Sepolia faucet; staking value plus gas must fit your balance.

Lab write-up

- Record your deployed address, the constructor rewardRate, and three tx hashes (stake, claim, unstake).
- Explain in two sentences why rewards must be pre-funded rather than minted.
- Note the gas used by stake() versus claimRewards() and say which storage writes explain the difference.