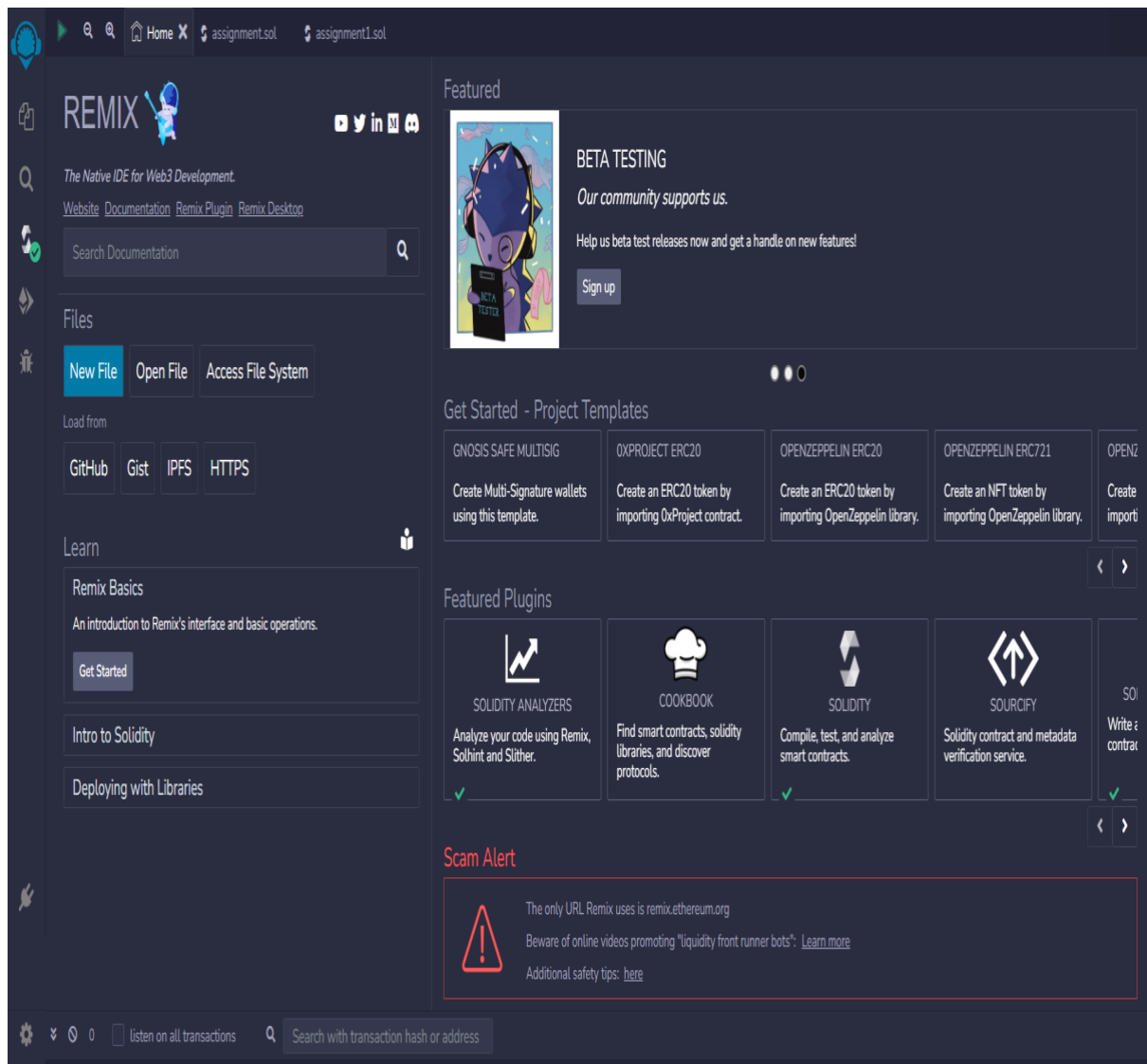


Assignment 1

NAME		ANGELIN BEULA. I
ZONE		4
COLLEGE		GKM college of engineering and technology

1. GO TO THE CHROME ATFORM OPEN REMIX PLAT FORM

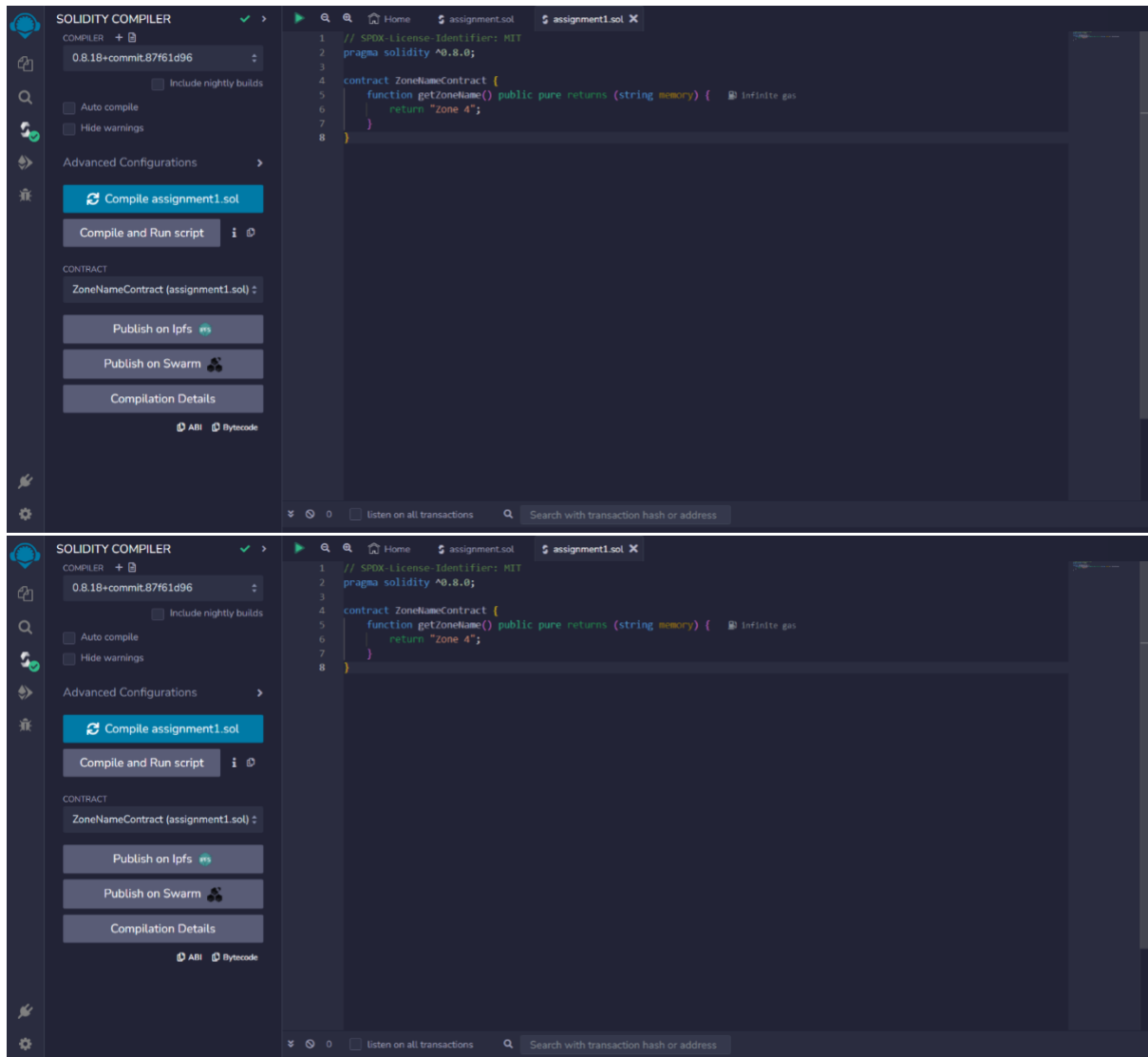


2. OPEN THE REMIX PAGE AND CREATE A NEW FILE

The screenshot displays the Remix IDE interface, which is a web-based development environment for Ethereum. The interface is divided into several panels:

- FILE EXPLORER (Left Panel):** Shows the workspace structure. It includes a "WORKSPACES" section with "default_workspace" and a "FILES" section with folders for "artifacts", "contracts", "scripts", and "source".
- Main Panel (Center):** Displays the "REMIX" logo and the tagline "The Native IDE for Web3 Development." Below this, there are links for "Website", "Documentation", "Remix Plugin", and "Remix Desktop". A search bar for "Search Documentation" is present. The "Files" section includes buttons for "New File", "Open File", and "Access File System". Under "Load from", there are buttons for "GitHub", "Gist", "IPFS", and "HTTPS". The "Learn" section includes links for "Remix Basics", "Intro to Solidity", and "Deploying with Libraries".
- Featured (Right Panel):** Contains a "WATCH TO LEARN" section with a video tip from the Remix Team. Below this is the "Get Started - Project Templates" section, which includes templates for "GNOSIS SAFE MULTISIG", "0XPROJECT ERC20", "OPENZEPPPELIN ERC20", and "OPENZEI". The "Featured Plugins" section includes "SOLIDITY ANALYZERS", "COOKBOOK", and "SOLIDITY".
- Scam Alert (Bottom Right):** A warning box stating "The only URL Remix uses is remix.ethereum.org" and "Beware of online videos promoting 'liquidity front runner bots'".
- Console (Bottom Panel):** Displays the "Welcome to Remix 0.36.3" message. It includes instructions on how to use the terminal, such as checking transactions, executing JavaScript scripts, and accessing libraries. The console also shows the results of a transaction, including the "from" address, "to" address, "value", "data", "logs", and "hash".

3. IN THE NEWLY CREATED FILE, CREATE A PROGRAM TO RETURN YOUR STRING, ” ZONE NAME ”



4. SAVE THE PROGRAM AND COMPILE IT TO GET THE ABI AND BYTECODE

ABI:

```
[
  {
    "inputs": [],
    "name": "getZoneName",
    "outputs": [
      {
        "internalType": "string",
        "name": "",
        "type": "string"
      }
    ],
    "stateMutability": "pure",
    "type": "function"
  }
]
```

BYTECODE:

```
608060405234801561001057600080FD5B50610173806100206000396000F3FE608060405234801561
001057600080FD5B506004361061002B5760003560E01C8063235ED70F14610030575B600080FD5B61
003861004E565B604051610045919061011B565B60405180910390F35B60606040518060400160405
280600681526020017F5A6F6E65203400000000000000000000000000000000000000000000000000
00815250905090565B600081519050919050565B600082825260208201905092915050565B60005B8
38110156100C55780820151818401526020810190506100AA565B60008484015250505050565B6000
601F19601F8301169050919050565B60006100ED8261008B565B6100F78185610096565B9350610107
8185602086016100A7565B610110816100D1565B840191505092915050565B6000602082019050818
103600083015261013581846100E2565B90509291505056FEA2646970667358221220B54F5DD4DA839
1CBF00288AEF1597C979F670B509601A34710DA7BF5137D64A164736F6C63430008120033
```

5. FINALLY DEPLOY IT TO DISPLAY THE OUTPUT

