## **Test plan document :**

## **Purpose**

The goal of this test plan is to make sure that the **JPP Compiler** works as expected when users:

* Write code with preprocessor directives.
* Use flags to control which parts of the code are included or excluded.
* View and copy the compiled/preprocessed output.

We want to confirm that the app behaves correctly, is easy to use, and doesn’t break under normal conditions.

## **Features to Test**

Here are the key things we’ll be testing:

1. **Code Editor**
   * User can type or paste code.
   * Code is handled properly.
2. **Flags Input**
   * Users can enter flags (e.g. DEBUG, PROD) manually.
   * Compiler should recognize flags and apply them to the code.
3. **Compile/Preprocess Button**
   * Clicking the button should show the preprocessed output.
   * Output should reflect which flags are active.
4. **Output Area**
   * Output should update correctly after compiling.
   * Syntax should match the expected preprocessed code.
   * Copy functionality should work.
5. **Error Handling**
   * If there's a syntax issue or unmatched directive, a proper message should be shown.

**Test Scenarios**

| **Test Case** | **What We’ll Do** | **What Should Happen** |
| --- | --- | --- |
| Enter simple code and compile | Type Correct Code | Output should include accordingly |
| Add a flag and compile | Use DEBUG in flag input, then compile | DEBUG-related code is included in output |
| Remove flag and compile | Remove DEBUG from input and compile again | DEBUG-related code is excluded from output |
| Use invalid syntax | Type #ifdef TEST without #endif | App should show an error or fail gracefully |
| Copy output | Compile code and click “Copy Output” | Output is copied to clipboard |

## **Test Environment**

| **What** | **Details** |
| --- | --- |
| Platform | Web (Vercel-hosted) |
| Browsers | Chrome, Firefox, Edge, Safari |
| Devices | Desktop, Laptop |
| Testing Type | Manual Testing |
| Tools Used | Browser DevTools, Clipboard check |

## **Test Flow (What We’ll Do)**

1. Open the app.
2. Paste different code examples into the editor.
3. Add or remove flags in the input box.
4. Click the “Compile” button and check the output.
5. Try edge cases .
6. Click “Copy Output” and test clipboard functionality.

## **Test Timeline**

| **Task** | **Time Estimate** |
| --- | --- |
| Writing Test Cases | 1 day |
| Manual Testing | 2 days |
| Bug Reporting & Retesting | 1–2 days |
| Final Review | 1 day |

## **People Involved**

| **Role** | **Name /Team** |
| --- | --- |
| Tester | Rohit Jaiswal |
| Developer | Aman Vyas |
| Reviewer/PM | Ram Sharma |
|  |  |

## **Done When (Exit Criteria)**

* All key test cases pass.
* Flags affect the output correctly.
* No broken behavior or UI bugs found.
* Output can be copied reliably.
* Any known issues are logged with clear notes.