**Program and Process Documentation for Number System Converter**

**1. Program Overview**

**Program Name. Number System Converter.**

**Technology. Flutter using Dart.**

**Purpose. This app lets users convert numbers between bases like binary, octal, decimal, and hexadecimal. It handles integer inputs and floating-point decimals too. Then it shows the results in the other bases through a simple interface that anyone can use.**

**2. Main Features**

**Users pick the input base from a dropdown. Options are binary, octal, decimal, or hexadecimal.**

**They type the number into the field. Decimal works with whole numbers or ones with decimals.**

**The conversion happens right away as they type. It switches to all the other systems and puts the outputs on screen.**

**Each result has a copy button. Tap it and it goes to the clipboard.**

**Input gets checked for the chosen base. If something is wrong, an error pops up.**

**Down at the bottom, there's a card with tips. It explains how to use it and what inputs work best.**

**3. Process Flow**

**1. App Initialization.**

**It kicks off with main. That runs the NumberConverterApp widget.**

**2. UI Rendering.**

**The home page shows up first. NumberConverterHomePage.**

**User picks the input base from the dropdown.**

**They put a number in the field.**

**3. Input Handling.**

**The field watches for changes.**

**Every time something updates, \_convertNumber gets called.**

**4. Conversion Logic.**

**Empty input clears everything out.**

**Decimal with a point uses \_convertFractional. That handles the float part to binary, octal, hex. Up to eight digits after the point.**

**For whole numbers, it parses and uses toRadixString in Dart for the bases.**

**Bad input for the base shows an error. Results get wiped.**

**5. Result Display.**

**Cards show the conversions for each system.**

**If there's a result, the copy button appears there.**

**6. User Actions.**

**Clear the field if needed.**

**Copy any result to clipboard.**

**4. Key Functions**

**\_convertNumber.**

**It checks the input and does the conversions for every base supported.**

**\_convertFractional. Takes a double value and base. Precision defaults to eight.**

**Turns decimal float to the base. Deals with whole and fractional bits.**

**\_parseToDecimal. Takes input string and fromBase.**

**Checks it and turns to integer for that base.**

**\_clearResults and \_clearResultsOnly.**

**Wipes the results and any errors.**

**\_buildResultCard.**

**Makes the widget for the result card. Includes the copy button.**

**5. Error Handling**

**Invalid Input.**

**If it does not fit the base pattern, error message comes up.**

**Floating-Point Support.**

**Only decimal takes floats. Others stick to integers.**

**6. User Interface**

**Material Design.**

**Uses those components for a clean modern vibe.**

**Responsive Layout.**

**Scrollable and fits different screens.**

**Visual Feedback.**

**Errors and snackbars tell about bad input or when copied.**

**7. Extensibility**

**Code is set up modular. Add more bases or stuff like negatives, scientific notation. Not much change needed.**

**8. Example Usage**

**1. Pick Decimal from dropdown.**

**2. Type 12.75 in the field.**

**3. See the binary, octal, hex versions. Fractions included.**

**4. Tap copy next to a result. It goes to clipboard.**