API Step-by-step workflow:

1. Health Check

* Endpoint: /
* Returns a simple JSON status message confirming the API is running

1. Variant Lookup by Gene + HGVS

* Endpoint: /user\_classify\_variant
* Inputs: gene, hgvs\_c, optional hgvs\_p
* Process:
* Connects to PostgreSQL
* Queries gene\_variants for a match
* Returns a structured JSON with:
* Variant ID
* Gene Symbol
* HGVS notations
* Protein Position
* Molecular Consequence
* Clinical Significance
* Review Status
* Other non-core fields
* Returns error if no match found

1. Apply ACMG Rules to All Variants

* Endpoint: /acmg\_classify\_all
* Loads all variants from the database into a Pandas DataFrame
* Passes them to apply\_ps1\_pm5\_pp5\_bp6 (from new2.py) – This applies ACMG support criteria to mark variants
* Returns the updated classification results

1. Search Variants by Protein Position

* Endpoint: /search\_by\_protein\_pos
* Inputs: gene, position (protein position integer)
* Returns all variants in the same protein position for that gene

1. Search Variants by Molecular Consequence

* Endpoint: /search\_by\_consequence
* Inputs: gene, consequence (e.g. missense\_variant)
* Returns all variants in the gene with that consequence type

1. Summarize Classification Counts

* Endpoint: /classification\_summary
* Counts the number of variants by classification (pathogenic, likely pathogenic, VUS, etc)
* Returns a frequency dictionary

Database

* Uses PostgreSQL (clinvar\_db)
* Table gene\_variants must already be populated (e.g. by your TP53 pipeline script)