**Phone Number Lookup and Validation Tool Project Report**

**Project Overview**

This project is a command-line-based phone number lookup and validation tool that combines local phone parsing (via the phonenumbers library) and external validation using IPQualityScore (IPQS) API. It provides comprehensive information about any phone number and allows optional CSV logging for record-keeping and analysis.

**Key Technologies**

* **Python 3**: Main programming language.
* **Phonenumbers**: For parsing, validating, formatting, and extracting metadata from phone numbers.
* **IPQualityScore (IPQS) API**: For external fraud detection, activity, risk, and metadata lookup.
* **CSV Module**: To save phone number details into a CSV file.
* **OS Module**: For file management (checking if the CSV exists).
* **Requests**: To handle API HTTP requests.

**Major Features**

**1. Local Phone Number Parsing (phonenumbers)**

* Parses any given number using phonenumbers.parse().
* Returns:
  + International formatted number (E.164 format).
  + Associated time zones.
  + Carrier/Network name.
  + Geographical region.
  + Country code.
  + Number type (mobile, fixed line, VoIP, etc.).
  + Whether the number is valid.
  + Whether the number is possible (structure-wise).

**2. External Validation (IPQS API Lookup)**

* Sends an HTTP GET request to IPQS with the number.
* Retrieves:
  + IPQS formatted and local versions of the number.
  + Validity (according to IPQS).
  + Fraud score (risk indicator).
  + Recent abuse reports.
  + Whether it’s a VoIP number.
  + Prepaid status.
  + Risk assessment.
  + Activity status.
  + Associated owner name (if available).
  + Line type classification.
  + Carrier, country, region, city, zip code, timezone, dialing code.
  + Flags for Do Not Call List, spam reports, and data leaks.

**3. User Interaction Loop**

* Prompts users to enter a phone number.
* Provides detailed printed information for each number.
* Optionally saves the lookup results to a CSV file.
* Supports graceful exit with the "exit" keyword.

**4. CSV Saving Functionality**

* Appends information about each lookup to a phone\_log.csv file.
* Automatically creates the CSV if it doesn’t exist.
* Proper UTF-8 encoding to support international characters.
* Ensures headers are only written once.

**Technical Details**

**IPQS API Integration**

* URL dynamically constructed using the user’s IPQS API key and input phone number.
* Handles network errors and JSON parsing issues gracefully.
* Prints specific error messages if the API request fails or returns an unsuccessful response.

**Error Handling**

* Catches number parsing errors (e.g., invalid input formats).
* Catches API connection errors and prints understandable alerts.
* Protects against file handling exceptions when saving to CSV.

**Formatting and Output**

* Clear sectioned printing with phone emojis and structured key-value format.
* Dynamic field handling based on available data.
* Case-insensitive input handling ("exit", "EXIT", "Exit" all recognized).

**Security Considerations**

* Sensitive data (API key) isolated at the top (IPQS\_API\_KEY) for easy replacement.
* Assumes that the user will replace YOUR\_API\_KEY\_HERE with their actual IPQS key.

**Code Structure**

* **ipqs\_lookup(phone\_number)**: Handles contacting IPQS and extracting meaningful fields.
* **get\_number\_info(mobileNo\_str)**: Handles local parsing + IPQS lookup + structured output.
* **save\_to\_csv(info, filename)**: Manages saving the lookup information into a persistent CSV.
* **Main Loop**: Ensures continuous operation until user decides to exit.

**Strengths**

* Combines local parsing and cloud-based verification.
* Robust error handling at all stages (network, input, file IO).
* Clean, organized, and readable code.
* Highly extendable (easy to add more fields or another API).

**Areas for Potential Improvement**

* **Input Validation**: Add a regex pre-check before parsing to catch obvious format mistakes.
* **Asynchronous API Requests**: Could speed up response times using asyncio and aiohttp.
* **Retry Mechanism**: Add retry logic on API request failures.
* **Logging**: Implement a logging system (e.g., logging module) instead of print statements for better diagnostics.
* **Config Management**: Externalize API key and settings into a config file or environment variables.
* **UI Enhancement**: Optionally integrate into a GUI (e.g., Tkinter or PyQt5).

**Conclusion**

The Phone Number Lookup and Validation Tool is a well-structured, powerful, and practical project combining local and external resources to provide deep insights into any phone number. Its clean code structure, strong error handling, and user-friendly interactions make it highly reliable for research, fraud prevention, or general contact validation tasks.