There were both easy parts and more difficult obstacles in developing a data tool to retrieve, convert, modify, and export data in various formats. Because each data format has subtleties, such as JSON orientation or SQL indexing requirements, supporting several data formats like CSV, JSON, and SQL required careful handling to ensure consistent parsing and conversion. We figured it was crucial to plan the controller flow to effectively handle these procedures, particularly when coordinating data retrieval, format conversion, and further changes. With the possibility of encountering various issues such as unsupported formats, missing files, or database connection problems, we also thought it was important to consider fault tolerance. The development of a strong error handling mechanism would guarantee that the application could continue or easily abort while also providing insightful feedback. Some parts were simpler than expected, even if these factors presented difficulties. Because the `requests` package made HTTP requests simple to handle, this made it easier to retrieve data from distant sources. Some jobs, though, proved more challenging. Complex data structures, especially nested JSON, required more processing during the conversion process in order to flatten the data while maintaining its meaning, which made fault tolerance and controller flow more difficult. It was also necessary for us to make several iterations to improve the data tool's flow and error resilience in order to provide code flexibility and reusability to handle different input and output types. This tool would be useful for future data science projects like automation of data transformation and cleaning—both crucial for getting data from various sources ready for analysis. It becomes more effective to combine data from databases, web APIs, and flat files into a single pipeline, particularly when fault-tolerant measures are in place to guarantee data workflow stability. With a controller flow built to manage data conversions with ease, the tool also facilitates the creation of complex ETL pipelines for processing massive amounts of data. Overall, creating this tool highlighted how crucial it is to create adaptable tools with strong controller flow and fault tolerance, increasing the effectiveness of data manipulation efforts.