TEST CASES FOR NATURAL LANGUAGE PROCESSING BASED COMMAND LINE APPLICATION

UNIT TESTING

Test Case ID	Component	Test Description	Input	Expected Output
UT-01	processCommand()	Should call sendToFlask() with user input	"create folder test"	Sends POST request to Flask, receives command
UT-02	askAndExecute()	Should execute command after user confirms	"yes" after prompt for mkdir test	Command executed, output displayed
UT-03	askAndExecute()	Should cancel command if user declines	"no"	"Command not executed." message shown
UT-04	sendToFlask()	Should send JSON data to Flask and return output	{ question, answer }	Interpreted command returned from Flask
UT-05	listCache()	Should list all Redis keys	-	Displays cached keys and values
UT-06	clearCache()	Should clear all Redis entries	-	Redis cleared message displayed

INTEGRATION TESTING

Test Case ID	Components Involved	Test Description	Input	Expected Output
IT-01	CLI + Flask	End-to-end POST request from CLI to backend	--command "list files"	Returns interpreted ls command
IT-02	Flask + Redis	Should store processed command in Redis	"mkdir test"	Cached key-value stored
IT-03	Redis + CLI	Should retrieve existing cache on repeated request	"mkdir test" (repeated)	Output fetched from cache
IT-04	Ollama + Flask + Transformer	LLM response improves command interpretation	"remove junk" + LLM context	Returns rm command

FUNCTIONAL TESTING

Test Case ID	Feature	Test Description	Input	Expected Output
FT-01	Command Translation	Interprets basic directory creation	"create folder named test"	mkdir test
FT-02	Error Handling	Handles model/server failure	Server down	Displays meaningful error message
FT-03	Cache Management	--list-cache displays cache entries	CLI flag --list-cache	Cached commands shown
FT-04	Confirmation Prompt	Prompts before executing command	"yes" to run command	Executes and shows output
FT-05	Invalid Input	Gracefully handles unrecognized input	"do something crazy"	Error or fallback message

VALIDATION TESTING

Test Case ID	User Focus Area	Test Description	Input	Expected Output
VT-01	Natural Input	Accepts flexible phrasings	"make a new directory test"	Returns mkdir test
VT-02	Safety Confirmation	Prevents unintended command execution	"delete all" + user says "no"	Command not executed
VT-03	Ease of Use	Works for users with no CLI experience	"show files"	Lists files (returns ls)
VT-04	Feedback & Output	Provides user-friendly output styling	Any command	Colored output via chalk

SYSTEM TESTING

Test Case ID	Entire System Component	Test Description	Scenario	Expected Output
ST-01	Full Stack	From input → NLP → Flask → Redis → exec → output	--command "list running processes"	Command executed + output shown
ST-02	Failure Recovery	Handles network/server failure gracefully	Flask server offline	Error message shown, CLI does not crash
ST-03	CLI + Redis + Backend	Validates multi-session cache behaviour	Run same command twice	Second run uses cache
ST-04	Render Deployment	Works with deployed backend and Redis	CLI calls Render-hosted Flask/Redis	Same performance and accuracy

ACCEPTANCE TESTING

Test Case ID	End-User Criteria	Test Description	Condition	Expected Outcome
AT-01	Usability	User can run the CLI and get meaningful results	User types "make dir x"	CLI shows mkdir x and executes
AT-02	Multi-language Intent	Understands varied phrasing and synonyms	"new folder x", "create dir x"	Both map to mkdir x
AT-03	Exit and Session Flow	Exits gracefully on "exit" in chat mode	User types "exit"	CLI thanks user and exits
AT-04	Stakeholder Review	Presented to mentor or supervisor	Full run-through	Approved for deployment