ISAD1000/5004 – ISE Assignment

Student: Ishan Renu Punj
Student Number: 21990726
Semester 2, 2025

Phase 1 – Setup

1. Git Repository

- **Repository Name:** IshanRenuPunj_21990726_ISE_Repo
- **Purpose:** Store all project code, test cases, and documentation with version control.

Commands Used:

- mkdir icsc
- cd icsc
- git init

2. README.md

First Commit:

- git add README
- git commit -m "add readme"

3. Branch Plan

Branch	Purpose
main	Stable version for submission
dev	Active development and integration of new features

testing	Implement and run black-box and white-box test cases
docs	Documentation, design notes, and final report

Branch Creation:

- git branch dev
- git branch testing
- git branch docs

Phase 2 – Design

1. Required Functions

Function Name	Purpose	Inputs	Outputs
load_services_from_file(path)	Load service details (thresholds, costs, units) from CSV file	path: str	dict of service data
find_tier_cost(amount, thresholds, costs)	Determine per-unit cost based on usage tier	amount: float, thresholds: list, costs: list	float
calculate_service_total(amount, service)	Calculate total cost for a service	amount: float, service: dict	float
display_service_structure(name, service)	Show pricing tiers for a service	name: str, service: dict	None (prints output)
list_subscriptions(subs, services)	Display all subscriptions and usage	subs: dict, services: dict	None (prints output)
show_breakdown(subs, services)	Show detailed cost breakdown per service and total	subs: dict, services: dict	None (prints output)
main()	Integrate modules and provide CLI interface	None	None (program flow)

2. Modularity Design Choices

- **Separation of Concerns:**
  - services_loader.py → Load CSV data.
  - calculator.py → Tier calculations & totals.
  - ui.py → Display menus, subscriptions, and breakdowns.
  - main.py → CLI interface and program flow.
- **Advantages:**
  - Easy maintenance and debugging.
  - Functions are reusable and testable independently.
  - Clear workflow allows incremental commits and better version control.

3. Sample services.csv

Compute,hour
0,50,1000,8000
0.62,0.58,0.55,0.52
Storage,Gb
0,100,500
0.12,0.10,0.09
Network,GB
0,1000,10000
0.09,0.07,0.05

4. Commit for Phase 2 Design

In dev branch -

- git add services.csv
- git commit -m "add sample services.csv"

In docs branch -

- git add docs/ISE_Assignment.pdf
- git commit –m “add initial design documentation”

Phase 3 – Implementation

1. ICSC Code Implementation

- Implemented all required functions for menu navigation, service selection, and cost calculations.
- Functions include: load_services_from_file(), find_tier_cost(), calculate_service_total(), display_service_structure(), list_subscriptions(), show_breakdown(), and main().
- Modular design from Phase 2 maintained for separation of concerns.

2. Loading services.csv and Handling User Input

- Services are loaded from services.csv at program start.
- Users can select services, enter subscription amounts, and modify previous entries.
- Input validation included for numeric values and valid service selection.

3. Cost Breakdown Display

- Added detailed display of cost per service and total monthly cost.
- Pricing tiers shown for transparency.
- Updates dynamically as user modifies subscription amounts.

4. Modularity Review and Refactoring

- Code reviewed for adherence to modularity principles.
- Functions tested independently to ensure reusability and correctness.
- Refactored minor issues for clarity and maintainability.

5. Version Control

- commit message for this phase:
    - "implement ICSC core functionality with modular design"