# FLEXI-INCOME

Income Stability Analysis & Salary Smoothing Eligibility


Presented By :

Louey Dridi
Dhie eddine Chedly
Ahmed Rejeb
Mohamed Belgacem

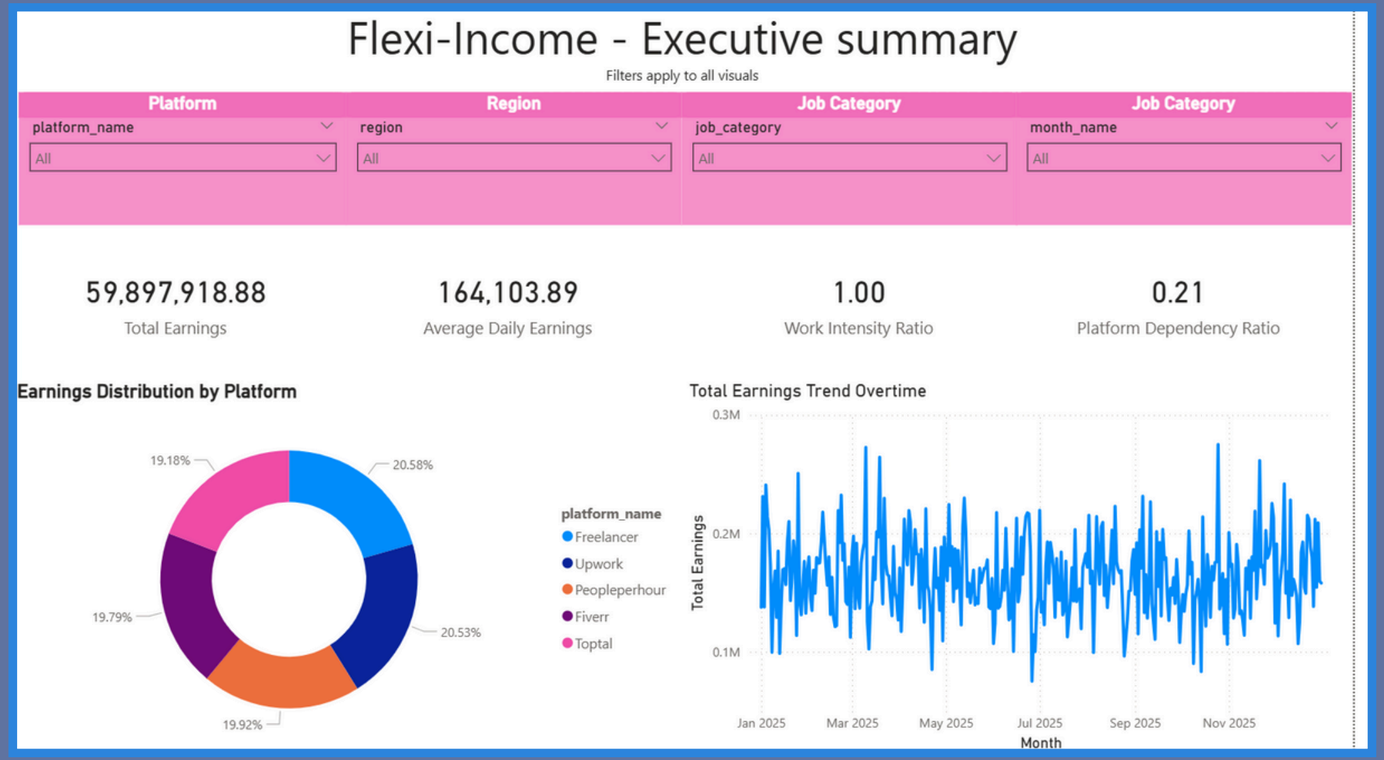
# Executive Summary

FlexiIncome

## Key takeaway:

High earnings volume does not imply income stability or repayment capacity

- Flexi-Income evaluates the feasibility of a Salary Smoothing product for gig workers
- Built using a dimensional BI model and interactive dashboards
- Portfolio exceeds $59M in total earnings
- However, income instability is widespread
- Only 5.59% of workers meet the strict volatility-based eligibility rule



# Business Context

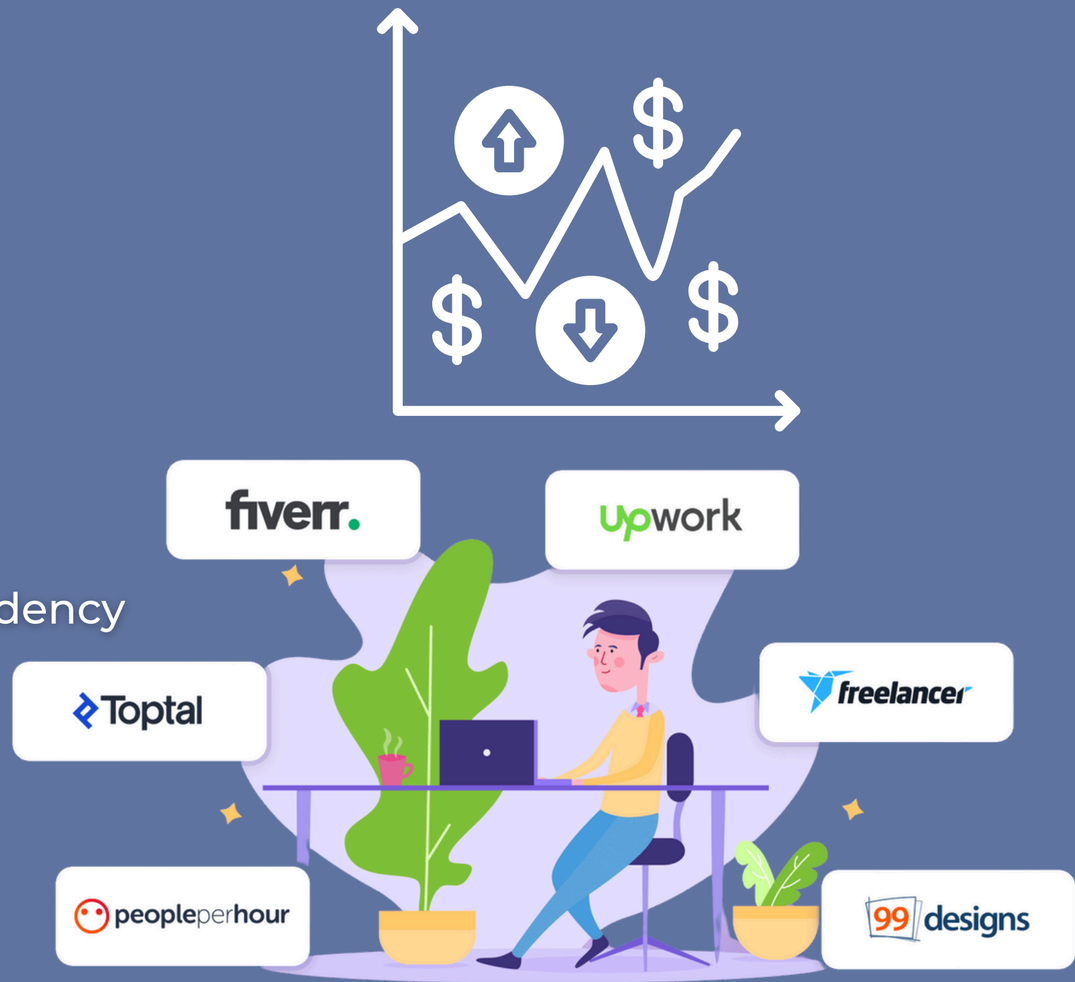
FlexiIncome

## Industry

- Fintech / Neobanking / Embedded Finance
- Rapid growth of the global gig economy

## Problem

- Traditional banks rely on fixed salaries
- Gig workers have irregular and volatile income
- Many are underbanked or excluded from credit products
- Earnings fluctuate due to demand, gaps, and platform dependency



# Organization & Business Problem

FlexiIncome

## Organization: GigFin

- Digital-first neobank for gig economy workers
- Builds financial profiles from transactional data

## Business Challenge:

- Frequent income volatility and gap periods
- Difficulty distinguishing:
- Structurally risky workers
- Temporarily unstable but viable workers
- Need for data-driven salary smoothing decisions

## The Solution

- Business Intelligence data mart
- Transforms raw gig-economy transactions into analytical insights
- Supports safer and more automated lending decisions

## Project Objectives

- Analyze income behavior at daily granularity
- Measure income stability and volatility
- Identify eligible workers for salary smoothing
- Support decision-making through dashboards

FlexiIncome

# Dataset Overview

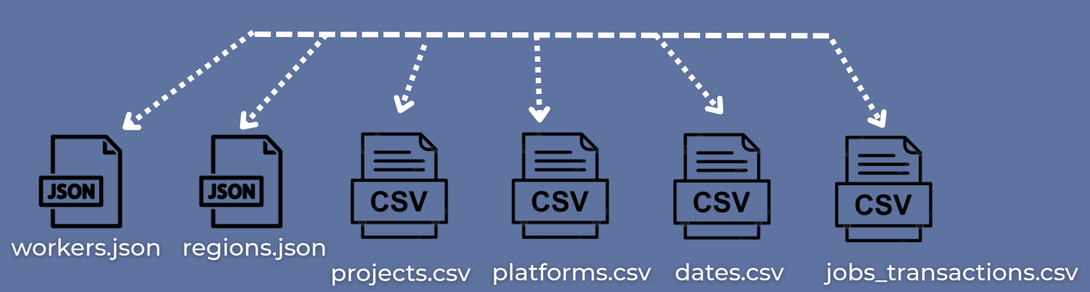
## Data Characteristics

- Daily earnings transactions
- Historical multi-month data
- Currency normalized to USD

## Main Entities

- Workers
- Platforms
- Dates
- Regions
- Projects

## Gathered Data:



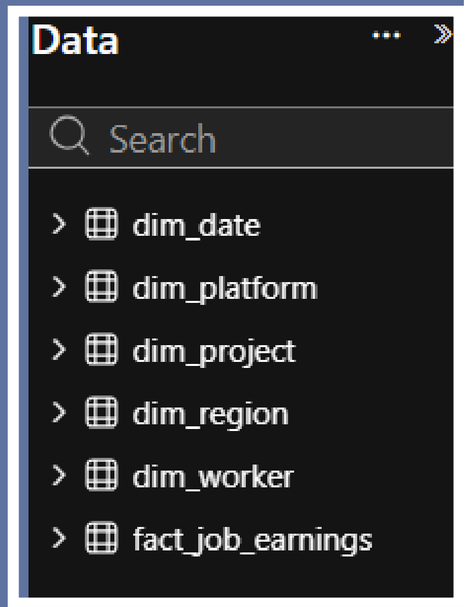

- data_cleaned
  - dim_date.csv
  - dim_platform.csv
  - dim_project.csv
  - dim_region.csv
  - dim_worker.csv
  - fact_job_earnings.csv
- data_raw
  - dates.csv
  - jobs_transactions.csv
  - platforms.csv
  - projects.csv
  - regions.json
  - workers.json

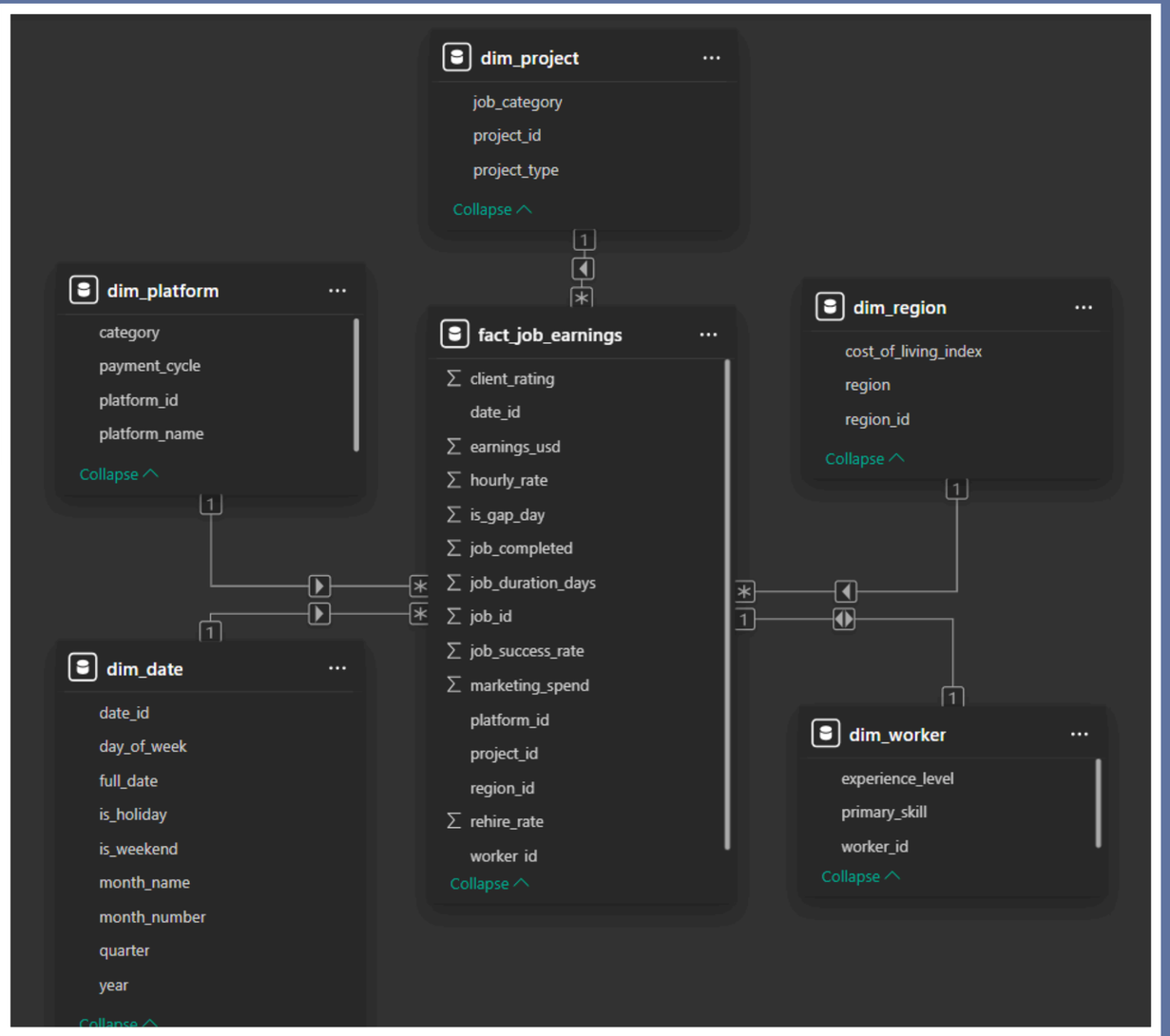
| job_id | worker_id | platform_id | region_id | project_id | date_id | earnings_usd | job_completed | job_duration_da | hourly_rate | job_success_rate | client_rating | rehire_rate | marketing_sp | is_gap_day |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 4058 | 796 | 5 | 5 | 8 | 20251019 | 4759.09 | 3 | 41 | 82.09 | 88.5 | 4.51 | 84.15 | 270 | 0 |
| 4659 | 251 | 2 | 4 | 16 | 20250310 | 4759.09 | 113 | 59 | 82.09 | 72.41 | 4.07 | 47.94 | 348 | 0 |
| 9400 | 1362 | 3 | 7 | 3 | 20251011 | 4759.09 | 7 | 44 | 82.09 | 74.98 | 4.45 | 39.29 | 58 | 0 |
| 3565 | 1037 | 2 | 7 | 13 | 20250603 | 4759.09 | 161 | 9 | 82.09 | 96.33 | 4.74 | 38.46 | 145 | 0 |
| 10256 | 1216 | 5 | 1 | 10 | 20250604 | 4759.09 | 86 | 10 | 82.09 | 84.46 | 4.47 | 54.82 | 263 | 0 |
| 12148 | 1209 | 3 | 4 | 6 | 20251207 | 4759.09 | 125 | 32 | 82.09 | 80.55 | 4.81 | 50.31 | 146 | 0 |

# Data Model

## Analytical Model

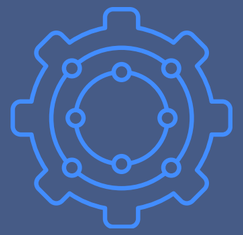
- Star schema design
- Central fact table: Job Earnings
- Dimension tables:
  - Worker
  - Platform
  - Date
  - Region
  - Project

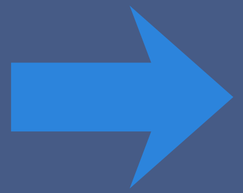


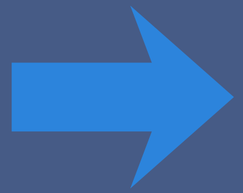



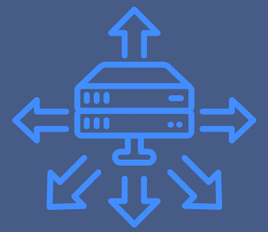
# ETL PROCESS

FlexiIncome









## Extraction

The automated ingestion of raw transactional data from external sources.

## Transformation

Cleaning, validating, and converting raw data into meaningful business metrics.

## Loading

Storing the processed data into a structured format for analysis.

```python
# Extract jobs transactions (main facts data)
jobs_df = pd.read_csv(RAW_DATA_DIR / 'jobs_transactions.csv')
print(f"Jobs transactions: {len(jobs_df)} records")
jobs_df.head()
```

```python
# Extract workers dimension (JSON)
with open(RAW_DATA_DIR / 'workers.json', 'r') as f:
    workers_raw = json.load(f)
workers_df = pd.DataFrame(workers_raw)
print(f"Workers: {len(workers_df)} records")
workers_df.head()
```

# Extracting

Using Python, we created a code that extracts data from different data types that will be used in the next steps

# Data Transformation

converting raw logs into risk intelligence

- **Currency Normalization**: All multi-currency earnings (GBP, EUR, AUD) converted to USD for a unified baseline.
- **Gap Detection Logic**: Algorithm identified days with $0 earnings to flag "Gap Days" (Risk Indicators).
- **Volatility Calculation**: Computed standard deviation of daily income to derive the Income Volatility Index (IVI).

| job_id | worker_id | platform_id | region_id | project_id | date_id | earnings_usd | job_completed | job_duration_days |
|---|---|---|---|---|---|---|---|---|
| 4058 | 796 | 5 | 5 | 8 | 20251019 | 4759.09 | 3 | 41 |
| 4659 | 251 | 2 | 4 | 16 | 20250310 | 4759.09 | 113 | 59 |
| 9400 | 1362 | 3 | 7 | 3 | 20251011 | 4759.09 | 7 | 44 |
| 3565 | 1037 | 2 | 7 | 13 | 20250603 | 4759.09 | 161 | 9 |
| 10256 | 1216 | 5 | 1 | 10 | 20250604 | 4759.09 | 86 | 10 |
| 12148 | 1209 | 3 | 4 | 6 | 20251207 | 4759.09 | 125 | 32 |
| 5640 | 1487 | 3 | 2 | 15 | 20250925 | 4759.09 | 198 | 35 |
| 312 | 1031 | 3 | 3 | 6 | 20251022 | 4759.09 | 125 | 29 |
| 5859 | 1859 | 4 | 5 | 7 | 20250201 | 4759.09 | 102 | 23 |
| 1022 | 255 | 4 | 7 | 3 | 20250407 | 4759.09 | 282 | 17 |
| 378 | 991 | 5 | 7 | 11 | 20250612 | 4759.09 | 67 | 37 |
| 7337 | 1936 | 1 | 7 | 16 | 20250406 | 4759.09 | 76 | 31 |
| 3669 | 1531 | 1 | 4 | 9 | 20251224 | 4759.09 | 178 | 27 |
| 10501 | 688 | 3 | 4 | 12 | 20250123 | 4759.09 | 56 | 46 |
| 2933 | 848 | 2 | 1 | 7 | 20250108 | 4759.09 | 172 | 27 |
| 8325 | 1281 | 3 | 2 | 8 | 20250223 | 4759.09 | 152 | 2 |
| 2797 | 651 | 4 | 5 | 16 | 20250404 | 4759.09 | 292 | 34 |
| 3711 | 1418 | 3 | 6 | 7 | 20250624 | 4759.09 | 149 | 60 |

# LOADING

## The Star Schema Data Model

```
# Save dimension tables
dim_worker.to_csv(CLEANED_DATA_DIR / 'dim_worker.csv', index=False)
print(f"✓ Saved dim_worker.csv ({len(dim_worker)} records)")

dim_platform.to_csv(CLEANED_DATA_DIR / 'dim_platform.csv', index=False)
print(f"✓ Saved dim_platform.csv ({len(dim_platform)} records)")

dim_region.to_csv(CLEANED_DATA_DIR / 'dim_region.csv', index=False)
print(f"✓ Saved dim_region.csv ({len(dim_region)} records)")

dim_project.to_csv(CLEANED_DATA_DIR / 'dim_project.csv', index=False)
print(f"✓ Saved dim_project.csv ({len(dim_project)} records)")

dim_date.to_csv(CLEANED_DATA_DIR / 'dim_date.csv', index=False)
print(f"✓ Saved dim_date.csv ({len(dim_date)} records)")
```

FlexiIncome

The Transformation Strategy Instead of analyzing one giant, messy file, we transformed the raw data into a Star Schema architecture to optimize performance and clarity

**We split descriptive data into 5 specific "Dimensions" to slice and dice the analytics:**

- **dim_worker:** Demographics & Risk Profiles.
- **dim_platform:** Source of income (Upwork, Fiverr, etc.).
- **dim_region:** Geographic & Cost of Living data.
- **dim_project:** Job categories (Design, Dev, etc.).
- **dim_date:** Calendar logic (identifying weekends/holidays).

**The Result (Fact Table)**

- **fact_job_earnings:** The central table containing the calculated metrics (Earnings, Volatility Scores, Gap Flags).

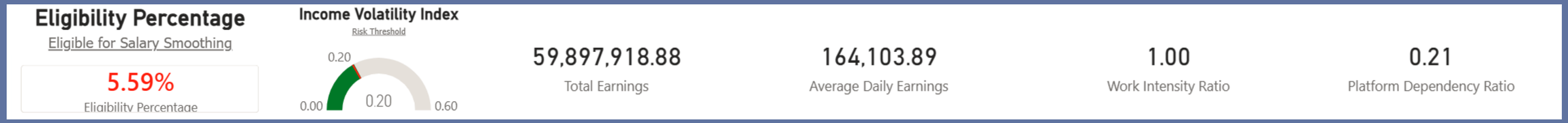
# Key Metrics Framework

FlexiIncome

## Core Metrics

- Total Earnings
- Average Daily Earnings (ADE)
- Income Volatility Index (IVI)
- Gap Day Frequency
- Average Gap Duration
- Platform Dependency Ratio
- Work Intensity Ratio

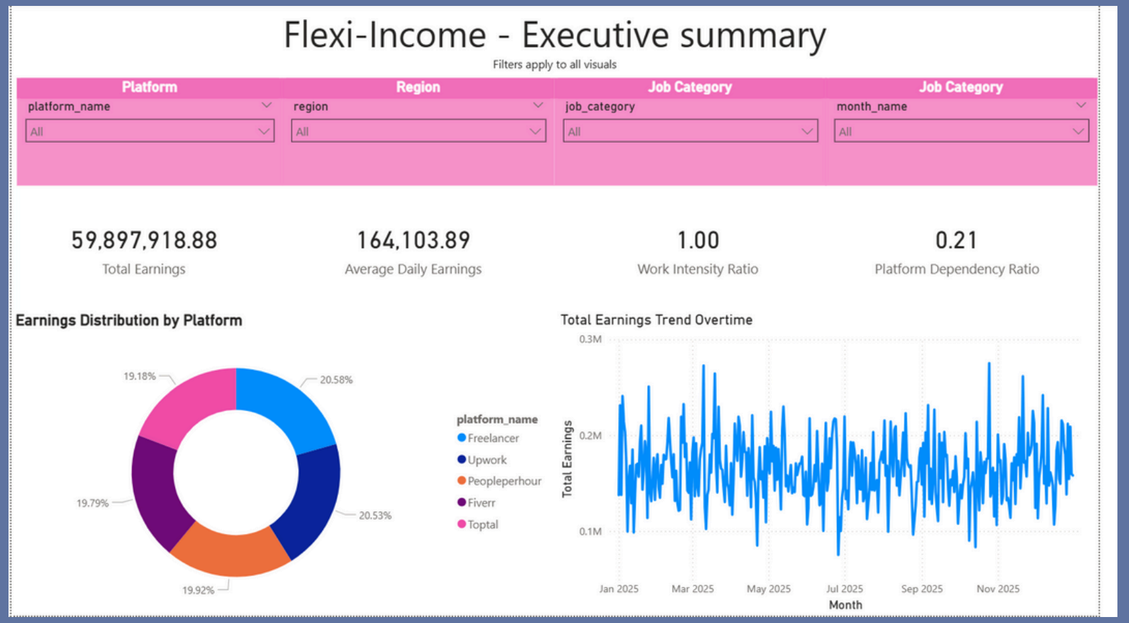
| KPI Name | Definition | Business Value |
|---|---|---|
| Avg Daily Earnings (ADE) | Total Earnings / Active Days | Establishes baseline income level. |
| Income Volatility Index (IVI) | $\sigma(DailyEarnings)/\mu(DailyEarnings)$ | Core risk metric for lending eligibility. |
| Gap Day Frequency | % of days with zero income | Identifies income reliability risk. |
| Platform Dependency Ratio | Top Platform Earnings / Total Earnings | Measures platform concentration risk. |
| Eligibility % | % of workers with IVI $< 20\%$ | Estimates product Total Addressable Market. |
| Avg Gap Duration | Avg consecutive zero-income days | Determines salary-smoothing loan duration. |
| WoW Growth | $(Inc_{Curr} - Inc_{Prev})/Inc_{Prev}$ | Tracks short-term income trends. |
| Work Intensity Ratio | Active Days / Total Days in Period | Differentiates full-time from casual workers and supports stability analysis. |

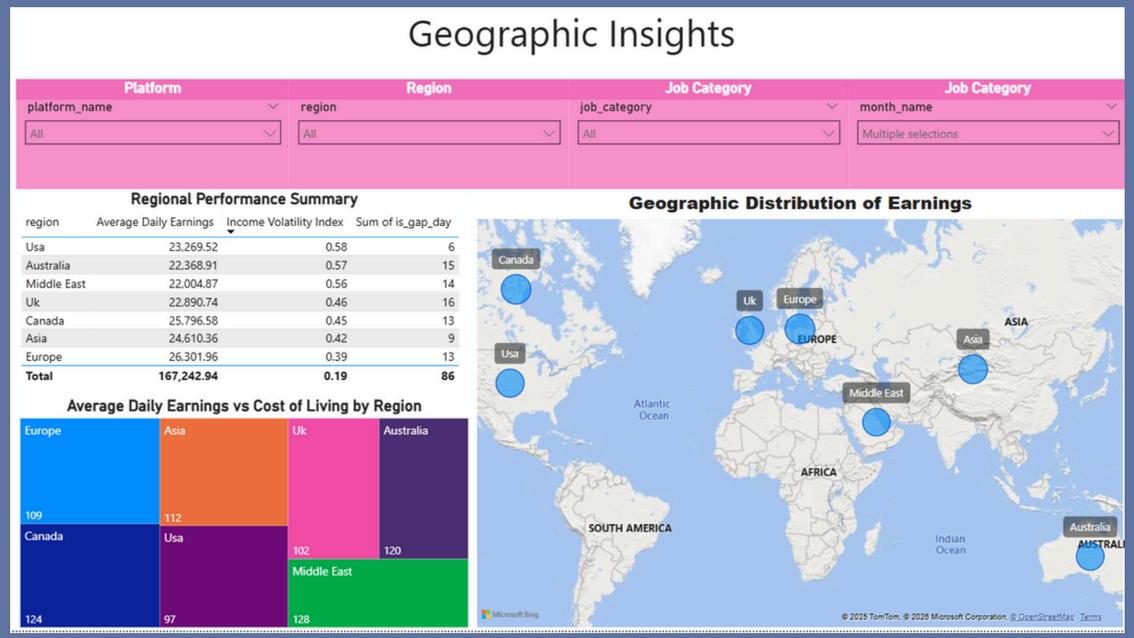


# Dashboard Overview

## Dashboard Features

- Worker stability profiles
- Eligibility tracking
- Volatility trends over time
- Regional and platform comparisons





FlexiIncome

FlexiIncome

# Key Insights

- Portfolio generates $59M+ in earnings, but high volume masks worker-level income instability
- Volatility is not platform-specific; risk is worker-driven, not platform-driven
- Only 5.59% of workers qualify under the strict IVI < 20% rule (eligibility paradox)
- Income gaps vary structurally by job category, making uniform risk rules ineffective
- Work consistency (high Work Intensity) is a stronger predictor of stability than income level
- Select regions combine lower volatility + favorable cost of living, ideal for initial rollout
- Geography matters: similar earnings can translate into very different real purchasing power

# Business Recommendations

FlexiIncome

- Implement Dual-Metric Eligibility Screening
- Adjust Risk Rules by Job Category
- Replace Hard Credit Freezes with Graduated Responses
- Prioritize Regional Rollout Based on Real Purchasing Power
- Maintain Platform-Neutral Credit Scoring
- Treat Salary Smoothing as a Learning Product

FlexiIncome

# Limitations 

- Absence of loan repayment and default data.
- Reliance on historical income patterns.
- Lack of expense and household financial data.
- Regional aggregation limits city-level insights.
- Certain historically stable job categories are highly exposed to automation by LLMs and AI agents which may cause future income stability to be overestimated.

# Future Improvements

FlexiIncome

- Integrate repayment and delinquency data.
- Introduce predictive income-forecasting models.
- Extend time horizons for seasonality analysis.
- Treat Salary Smoothing as a Learning Product
- Incorporate an AI exposure score by job category to adjust long-term income stability assessment.

# THANK YOU

FOR YOUR ATTENTION!