**Cursor Prompts for Income Comparison Feature**

**Context for All Prompts**

**Project**: Mingus resume AI feature for African American professionals aged 25-35 earning $40k-100k **Goal**: Add income comparison functionality that shows users how their salary compares to demographic peers **Target**: Motivate career advancement by showing specific income gaps and opportunities **Tech Stack**: Python, Flask, free APIs, ultra-budget MVP approach

**Prompt 1: Core Income Comparator Service**

Create a comprehensive IncomeComparator class for analyzing user income against demographic benchmarks.

Context: This is for Mingus, a financial app helping African American professionals (ages 25-35, $40k-100k) find better-paying jobs. Users need to see exactly where they stand financially compared to their demographic peers to motivate career advancement.

Requirements:

1. Compare user income to multiple demographic groups: national median, racial demographic, age group, education level, geographic location

2. Use 2022 American Community Survey data as fallback (hardcoded for reliability)

3. Calculate percentile rankings and income gaps

4. Generate motivational insights and action items

5. Handle missing data gracefully with fallback values

6. Focus on target metro areas: Atlanta, Houston, DC, Dallas, NYC, Philadelphia, Chicago, Charlotte, Miami, Baltimore

Key features needed:

- IncomeComparison dataclass with user\_income, comparison\_group, median\_income, percentile\_rank, income\_gap, gap\_percentage, context\_message

- Multiple comparison methods (national, racial, age, education, location)

- Percentile calculation algorithm

- Motivational insight generation for career advancement

- Integration points for Census Bureau API (future enhancement)

- Fallback data for African American income statistics by metro area

The class should emphasize financial opportunity and career growth, not just statistics. Include specific dollar amounts and percentage improvements that connect to job search motivation.

Include comprehensive error handling, logging, and example usage that demonstrates the full comparison pipeline.

**Prompt 2: Flask App Integration**

Modify the existing Flask application to integrate income comparison functionality with the resume analysis workflow.

Context: Add income comparison to the Mingus resume AI feature. Users upload resumes and provide demographic info, then see both job matches AND how their income compares to peers. This motivates career advancement by showing specific income gaps.

Modify these existing routes:

1. Update upload form route to collect demographic information (age, race, education, location, current salary)

2. Enhance the resume processing route to include income comparison analysis

3. Pass income comparison data to results template

4. Add error handling for demographic analysis failures

5. Maintain existing resume parsing and job matching functionality

New form fields needed:

- current\_salary (integer, optional but recommended)

- age (select dropdown for age ranges 25-27, 28-30, 31-33, 34-36, 37-40)

- race (dropdown with options including African American, White, Hispanic/Latino, Asian, Other)

- education (dropdown: High School, Some College, Bachelor's, Master's, Doctoral)

- location (existing metro area dropdown)

Integration requirements:

- Import and initialize IncomeComparator class

- Call income comparison when salary is provided

- Pass results to template: income\_comparisons list and motivational\_insights dict

- Handle cases where demographic data is incomplete

- Maintain backward compatibility with existing functionality

The integration should feel seamless - income comparison enhances the existing resume analysis rather than replacing it. Users should see both job opportunities AND their current financial position context.

Include proper error handling and graceful degradation when income comparison fails.

**Prompt 3: Enhanced User Input Form**

Create an enhanced HTML form template for collecting demographic information for income comparison analysis.

Context: Users of Mingus (African American professionals seeking career advancement) need to provide demographic info to see how their income compares to peers. The form should encourage participation while being respectful of privacy concerns.

Enhance the existing upload.html template with:

1. New section for "Income Analysis" that explains the value proposition

2. Form fields for demographic information (salary, age, race, education)

3. Progressive disclosure - make demographic fields optional but encouraged

4. Clear privacy messaging about how data is used

5. Professional, trustworthy design appropriate for career services

6. Mobile-responsive form layout

7. Input validation and user-friendly error messaging

Key design principles:

- Position income analysis as valuable career insight, not just data collection

- Use encouraging language about career advancement and financial growth

- Make salary input feel safe and private

- Provide clear value proposition: "See how your income compares and find opportunities"

- Include trust indicators and privacy assurances

Form enhancements needed:

- Section header explaining income comparison benefits

- Salary input with placeholder showing example values

- Age dropdown with ranges relevant to target demographic

- Race/ethnicity dropdown with inclusive options

- Education level dropdown covering typical career stages

- Enhanced location dropdown for target metro areas

- Submit button text updated to reflect new functionality

The form should feel like a career assessment tool rather than a data collection form. Include helpful text that explains how each field helps provide better career recommendations.

Style with professional CSS that matches the existing Mingus brand aesthetic - trustworthy, modern, and focused on financial growth.

**Prompt 4: Results Display Template**

Create a comprehensive results template that displays income comparison analysis alongside existing job matches.

Context: Mingus users need to see both their income position relative to demographic peers AND job opportunities that can improve their financial situation. This creates motivation for career advancement by showing specific income gaps and opportunities.

Create enhanced results.html template sections:

1. Income Analysis dashboard showing multiple demographic comparisons

2. Visual indicators for percentile rankings and income gaps

3. Motivational messaging about career advancement opportunities

4. Integration with existing job matches to show financial impact

5. Action items and next steps for income improvement

6. Professional design that builds confidence in career decisions

Key visual components:

- Comparison cards for each demographic group (national, racial, age, education, location)

- Percentile ranking displays with visual indicators

- Income gap highlights with dollar amounts and percentages

- Opportunity callouts showing potential income increases

- Integration between income analysis and job recommendations

- Motivational insights section with specific action items

Design requirements:

- Professional color scheme that conveys trust and financial growth

- Cards/grid layout for easy comparison scanning

- Clear typography hierarchy for financial data

- Responsive design for mobile users

- Positive, motivational tone throughout

- Visual emphasis on opportunities rather than just gaps

Each comparison should show:

- User's current income vs group median

- Percentile ranking with visual representation

- Context message explaining the comparison

- Opportunity highlight if significant gap exists

- Connection to available job matches when relevant

Include CSS styling that makes financial data feel empowering rather than discouraging. Use colors and visual cues that emphasize growth potential and career advancement opportunities.

The template should work seamlessly with existing job match results, creating a comprehensive career advancement dashboard.

**Prompt 5: Data Management and Fallback System**

Create a robust data management system for income comparison with fallback data and future API integration capabilities.

Context: Mingus needs reliable income comparison data for African American professionals across target metro areas. The system should work offline with fallback data but be ready for Census Bureau API integration.

Create comprehensive data management:

1. Fallback dataset with 2022 ACS income data by demographic groups

2. Data structure for easy updates and maintenance

3. API integration framework for Census Bureau API (future)

4. Data validation and quality checks

5. Geographic mapping for metro area income data

6. Error handling for missing or invalid data

Specific data needed:

- National median income by demographic groups

- African American median income by age groups (25-34, 35-44)

- Education level income medians by race

- Metro area income medians for target cities

- Income distribution data for percentile calculations

Data structure requirements:

- JSON/dict format for easy access and updates

- Hierarchical organization (national -> racial -> geographic -> age/education)

- Version tracking for data updates

- Documentation of data sources and update frequencies

- Error codes for missing data scenarios

API integration framework:

- Census Bureau API client class

- Rate limiting and error handling

- Data caching to minimize API calls

- Graceful fallback to offline data

- Data freshness validation

The system should prioritize reliability over real-time data - career decisions need consistent, trustworthy information. Include comprehensive logging and monitoring for data quality issues.

Create update procedures for refreshing fallback data annually and validation scripts to ensure data integrity across all demographic categories.

**Prompt 6: Testing and Validation Suite**

Create comprehensive testing suite for the income comparison feature including unit tests, integration tests, and sample demographic scenarios.

Context: Income comparison affects users' career decisions and financial planning, so reliability is critical. Test all demographic combinations relevant to Mingus target users.

Create testing coverage for:

1. IncomeComparator class methods and calculations

2. Demographic data accuracy and completeness

3. Percentile calculation validation

4. Error handling for edge cases

5. Flask integration testing

6. Template rendering with various data scenarios

7. Performance testing for comparison calculations

Specific test scenarios:

- User income comparisons across all target demographics

- Edge cases: very high/low incomes, missing demographic data

- Geographic variations across target metro areas

- Education level impact on income comparisons

- Age group transitions and income expectations

- API integration failure scenarios

Test data requirements:

- Sample user profiles representing target demographic

- Mock income data for consistent testing

- Edge case scenarios (outlier incomes, missing data)

- Geographic test cases for all target metro areas

- Educational background variations

Validation criteria:

- Percentile calculations match expected statistical distributions

- Income gaps calculated accurately with proper formatting

- Motivational messaging appropriate for different income levels

- Error handling graceful across all failure modes

- Performance acceptable for web application usage

Include integration tests that verify the complete user journey: form submission -> demographic analysis -> results display -> job match integration.

Create sample test users with realistic profiles from target demographic and validate that all comparisons and insights are accurate and motivational.

The testing should ensure users receive reliable, encouraging financial analysis that motivates career advancement rather than discourages job searching.

**Prompt 7: UI/UX Enhancements and Styling**

Create professional UI/UX design and CSS styling for the income comparison feature that motivates career advancement.

Context: Mingus serves African American professionals who need to feel confident about their career decisions. The income comparison UI should be empowering, professional, and visually compelling while maintaining trust and privacy.

Design comprehensive styling for:

1. Enhanced upload form with demographic collection

2. Income comparison dashboard with multiple comparison cards

3. Visual elements for percentile rankings and income gaps

4. Motivational design elements that encourage career growth

5. Integration styling that connects income analysis to job matches

6. Responsive design for mobile-first users

7. Accessibility compliance for inclusive design

Key design principles:

- Professional aesthetic appropriate for career services

- Color psychology: greens for growth, blues for trust, avoid reds for financial data

- Clear visual hierarchy emphasizing opportunities over gaps

- Mobile-first responsive design

- Trust indicators and privacy assurance styling

- Motivational visual cues and progress indicators

Specific UI components needed:

- Form section styling for demographic collection

- Comparison card layouts with financial data visualization

- Percentile ranking visual indicators (progress bars, badges, charts)

- Income gap highlighting with positive framing

- Opportunity callout boxes with actionable next steps

- Integration styling between income analysis and job recommendations

Visual enhancements:

- Gradient backgrounds suggesting upward financial trajectory

- Card-based layouts for easy information scanning

- Typography hierarchy that emphasizes key financial insights

- Loading states for comparison calculations

- Success states showing positive income comparisons

- Hover effects and micro-interactions for engagement

Accessibility requirements:

- WCAG 2.1 AA compliance for screen readers

- High contrast ratios for financial data readability

- Keyboard navigation support

- Alternative text for data visualizations

- Focus indicators for form interactions

The overall design should make users feel optimistic about their financial future while providing clear, actionable information about career advancement opportunities. Avoid any design elements that could feel judgmental or discouraging about current income levels.

Include CSS animations and transitions that suggest upward movement and progress, reinforcing the career advancement messaging.

**Prompt 8: Performance Optimization and Deployment**

Optimize the income comparison feature for production deployment with focus on performance, scalability, and cost-effectiveness.

Context: Mingus operates on ultra-budget constraints while serving users who need fast, reliable career analysis. The income comparison feature should perform well under load while minimizing operational costs.

Optimization requirements:

1. Efficient data structures and algorithms for income comparisons

2. Caching strategies for demographic data and API responses

3. Database optimization for user demographic storage (if needed)

4. API rate limiting and request optimization

5. Frontend performance optimization for data visualization

6. Memory usage optimization for demographic calculations

7. Deployment configuration for various hosting platforms

Performance targets:

- Income comparison calculations < 500ms

- Form submission to results display < 3 seconds

- Minimal memory footprint for demographic data

- Efficient API usage staying within free tier limits

- Mobile performance optimization for target users

Caching implementation:

- In-memory caching for frequently accessed demographic data

- Redis integration for session-based comparison caching

- API response caching with appropriate TTL values

- Static asset optimization for faster page loads

Deployment optimizations:

- Environment-specific configuration for dev/staging/production

- Monitoring and logging for performance tracking

- Error tracking for demographic analysis failures

- Cost monitoring for API usage and hosting resources

- Scalability considerations for increased user load

Database considerations:

- Optional user demographic storage for return visits

- Privacy-compliant data handling and retention policies

- Efficient indexing for demographic queries

- Data backup and recovery procedures

Security enhancements:

- Input validation for demographic data

- Rate limiting for form submissions

- HTTPS enforcement for sensitive financial data

- Data encryption for stored demographic information

The optimization should maintain the ultra-budget approach while ensuring professional-grade performance and reliability for users making important career decisions.

Include monitoring setup to track feature usage, performance metrics, and user engagement with income comparison analysis.

**Usage Instructions**

**Implementation Order:**

1. **Start with Prompt 1** - Core service foundation
2. **Prompt 5** - Data management (needed for testing)
3. **Prompt 2** - Flask integration
4. **Prompt 3** - Enhanced form
5. **Prompt 4** - Results display
6. **Prompt 6** - Testing suite
7. **Prompt 7** - UI/UX polish
8. **Prompt 8** - Production optimization

**Cursor-Specific Commands:**

# Reference existing code when integrating

@app.py Add income comparison integration to existing resume upload route

# Cross-reference files for consistency

@resume\_parser.py @income\_comparator.py Ensure data structures are compatible

# Optimize generated code

Optimize this income comparison algorithm for better performance

# Add error handling

Add comprehensive error handling to the demographic analysis workflow

**Validation Steps:**

1. **Test with sample data** after each prompt
2. **Verify integration** with existing resume analysis
3. **Check mobile responsiveness** for target users
4. **Validate calculations** with known demographic data
5. **Test error scenarios** and edge cases

**Expected Timeline:**

* **Day 1**: Prompts 1-2 (Core service + Flask integration)
* **Day 2**: Prompts 3-4 (Form + Results display)
* **Day 3**: Prompts 5-6 (Data management + Testing)
* **Day 4**: Prompts 7-8 (UI polish + Production ready)

Each prompt builds on the previous ones and creates a comprehensive income comparison system that motivates career advancement for your target demographic.