**Targeted Income-Boosting Job Recommendation Process**

**Process Overview**

**Goal**: Analyze resume → Compare income position → Find 3 targeted jobs in user's MSA and field that offer 15-30% salary increases

**Target Users**: African American professionals, ages 25-35, $40k-100k, in target metro areas

**Step-by-Step Process Flow**

**Phase 1: Enhanced Resume Analysis (5-10 minutes)**

**Step 1A: Document Processing & Skill Extraction**

# Enhanced resume parsing with field detection

resume\_data = {

'skills': ['Python', 'Project Management', 'SQL'],

'experience\_level': 'Mid-level (3-5 years)',

'industry': 'Technology',

'job\_titles': ['Business Analyst', 'Data Analyst'],

'education': 'Bachelor\'s Degree',

'certifications': ['PMP', 'Google Analytics'],

'current\_role\_seniority': 'Analyst',

'technical\_vs\_business\_split': 70, # 70% technical skills

'leadership\_indicators': ['team lead', 'project coordinator']

}

**Step 1B: Field of Expertise Classification**

# Determine primary and secondary fields

field\_analysis = {

'primary\_field': 'Data Analysis',

'secondary\_field': 'Project Management',

'industry\_focus': 'Technology/Finance',

'career\_trajectory': 'Individual Contributor → Team Lead',

'transferable\_skills': ['Communication', 'Problem Solving'],

'growth\_potential\_skills': ['Python', 'SQL', 'Tableau']

}

**Phase 2: Income Context Analysis (2-3 minutes)**

**Step 2A: Multi-Dimensional Income Comparison**

# Enhanced income analysis

income\_context = {

'user\_salary': 55000,

'percentile\_rankings': {

'national\_all\_workers': 45, # 45th percentile

'african\_american\_workers': 68, # 68th percentile

'atlanta\_metro\_area': 42, # 42nd percentile

'data\_analysts\_atlanta': 35, # 35th percentile - KEY INSIGHT

'college\_grads\_25\_35': 38 # 38th percentile

},

'income\_gaps': {

'field\_median\_gap': 12000, # $12k below data analyst median

'metro\_gap': 8000, # $8k below Atlanta median

'potential\_increase': 15000 # Target increase for 75th percentile

},

'opportunity\_score': 8.5 # Out of 10 - high opportunity for growth

}

**Step 2B: Target Income Calculation**

# Calculate optimal target salary range

target\_analysis = {

'minimum\_target': 65000, # 18% increase - conservative

'optimal\_target': 71000, # 29% increase - aggressive but achievable

'stretch\_target': 78000, # 42% increase - reach goal

'percentile\_goals': {

65000: '60th percentile in field',

71000: '75th percentile in field',

78000: '85th percentile in field'

}

}

**Phase 3: Intelligent Job Search & Filtering (8-12 minutes)**

**Step 3A: Multi-Source Job Aggregation**

# Search strategy across multiple APIs

job\_search\_strategy = {

'primary\_keywords': ['Data Analyst', 'Business Analyst', 'Python'],

'secondary\_keywords': ['SQL', 'Tableau', 'Project Management'],

'location\_filters': [

'Atlanta, GA',

'Atlanta Metro Area',

'Remote (Atlanta-based companies)',

'Hybrid Atlanta'

],

'salary\_filters': {

'minimum': 65000, # 18% increase minimum

'preferred': 71000 # 29% target

},

'company\_size\_preference': ['Medium', 'Large'], # Better benefits

'industry\_focus': ['Technology', 'Finance', 'Healthcare', 'Consulting']

}

**Step 3B: Advanced Job Scoring Algorithm**

# Comprehensive job scoring system

def calculate\_job\_score(job, user\_profile):

scores = {

'salary\_improvement': calculate\_salary\_score(job.salary, user\_profile.current\_salary),

'skills\_match': calculate\_skills\_alignment(job.description, user\_profile.skills),

'career\_progression': calculate\_growth\_potential(job.title, user\_profile.current\_level),

'company\_quality': calculate\_company\_score(job.company, job.benefits),

'location\_compatibility': calculate\_location\_score(job.location, user\_profile.metro),

'industry\_alignment': calculate\_industry\_fit(job.industry, user\_profile.background)

}

# Weighted scoring prioritizing income improvement

weighted\_score = (

scores['salary\_improvement'] \* 0.35 + # Income boost priority

scores['skills\_match'] \* 0.25 + # Qualification fit

scores['career\_progression'] \* 0.20 + # Growth potential

scores['company\_quality'] \* 0.10 + # Employer quality

scores['location\_compatibility'] \* 0.05 + # Geographic fit

scores['industry\_alignment'] \* 0.05 # Industry match

)

return weighted\_score

**Phase 4: Targeted Job Selection (3-5 minutes)**

**Step 4A: Three-Tier Job Selection Strategy**

# Select optimal mix of opportunities

job\_selection\_criteria = {

'conservative\_choice': {

'salary\_increase': '15-20%',

'skills\_match': '80%+',

'risk\_level': 'Low',

'description': 'Safe step up with proven skill alignment'

},

'optimal\_choice': {

'salary\_increase': '25-30%',

'skills\_match': '70%+',

'risk\_level': 'Medium',

'description': 'Strong growth opportunity requiring some skill development'

},

'stretch\_choice': {

'salary\_increase': '35%+',

'skills\_match': '60%+',

'risk\_level': 'Higher',

'description': 'High-reward opportunity for ambitious career leap'

}

}

**Step 4B: Final Recommendation Package**

# Complete recommendation with action plan

final\_recommendations = {

'job\_1\_conservative': {

'title': 'Senior Business Analyst',

'company': 'Delta Air Lines',

'salary\_range': '$62,000 - $68,000',

'salary\_increase': '18%',

'match\_score': 92,

'key\_requirements': ['SQL', 'Excel', 'Project Management'],

'missing\_skills': [],

'why\_selected': 'Strong skill alignment, stable company, clear path to growth',

'application\_strategy': 'Apply immediately - high probability of interview',

'income\_impact': 'Would move you to 58th percentile in your field'

},

'job\_2\_optimal': {

'title': 'Data Analyst II',

'company': 'Chick-fil-A Corporate',

'salary\_range': '$68,000 - $75,000',

'salary\_increase': '27%',

'match\_score': 88,

'key\_requirements': ['Python', 'SQL', 'Tableau', 'Statistics'],

'missing\_skills': ['Tableau (basic knowledge helpful)'],

'why\_selected': 'Growth company, excellent benefits, matches career trajectory',

'application\_strategy': 'Complete Tableau basics course, then apply',

'income\_impact': 'Would move you to 72nd percentile in your field'

},

'job\_3\_stretch': {

'title': 'Product Analyst',

'company': 'Mailchimp (Intuit)',

'salary\_range': '$75,000 - $85,000',

'salary\_increase': '42%',

'match\_score': 78,

'key\_requirements': ['Python', 'A/B Testing', 'Product Analytics', 'Stakeholder Management'],

'missing\_skills': ['A/B Testing', 'Product Analytics'],

'why\_selected': 'Tech company, high growth potential, strong compensation',

'application\_strategy': 'Network with current employees, take product analytics course',

'income\_impact': 'Would move you to 85th percentile in your field'

}

}

**Technical Implementation Steps**

**Step 1: Enhanced Resume Parser Class**

class AdvancedResumeParser(ResumeParser):

def \_\_init\_\_(self):

super().\_\_init\_\_()

self.field\_keywords = {

'data\_analysis': ['sql', 'python', 'tableau', 'excel', 'statistics', 'analytics'],

'project\_management': ['pmp', 'agile', 'scrum', 'stakeholder', 'timeline'],

'software\_development': ['javascript', 'react', 'node.js', 'api', 'git'],

'marketing': ['campaigns', 'social media', 'google analytics', 'seo'],

'finance': ['financial modeling', 'accounting', 'budgeting', 'forecasting'],

'sales': ['crm', 'salesforce', 'lead generation', 'client relations']

}

self.seniority\_indicators = {

'entry': ['intern', 'junior', 'assistant', 'associate'],

'mid': ['analyst', 'specialist', 'coordinator', 'senior'],

'senior': ['manager', 'director', 'lead', 'principal', 'head']

}

def analyze\_field\_expertise(self, resume\_data):

"""Determine primary field and expertise level"""

# Implementation here

pass

def calculate\_experience\_level(self, experience\_list):

"""Determine career progression level"""

# Implementation here

pass

def identify\_growth\_trajectory(self, job\_titles, skills):

"""Predict career advancement path"""

# Implementation here

pass

**Step 2: Income-Aware Job Matcher**

class IncomeBoostJobMatcher:

def \_\_init\_\_(self, income\_comparator, job\_searcher):

self.income\_comparator = income\_comparator

self.job\_searcher = job\_searcher

def find\_targeted\_opportunities(self, resume\_data, demographic\_data, income\_context):

"""Find 3 optimal jobs for income advancement"""

# 1. Define search parameters based on income goals

search\_params = self.\_calculate\_search\_parameters(income\_context)

# 2. Multi-source job search

all\_jobs = self.\_comprehensive\_job\_search(resume\_data, search\_params)

# 3. Score and filter jobs

scored\_jobs = self.\_score\_jobs\_for\_income\_boost(all\_jobs, resume\_data, income\_context)

# 4. Select diverse recommendations

final\_recommendations = self.\_select\_three\_tier\_recommendations(scored\_jobs)

return final\_recommendations

def \_calculate\_search\_parameters(self, income\_context):

"""Define job search criteria based on income analysis"""

current\_salary = income\_context['user\_salary']

target\_increase = 0.25 # 25% target increase

return {

'min\_salary': int(current\_salary \* 1.15), # Minimum 15% increase

'target\_salary': int(current\_salary \* (1 + target\_increase)),

'max\_salary': int(current\_salary \* 1.5), # Up to 50% stretch

'location\_radius': 25, # Miles from metro center

'remote\_acceptable': True

}

def \_comprehensive\_job\_search(self, resume\_data, search\_params):

"""Search multiple job sources with intelligent keywords"""

primary\_field = resume\_data['field\_analysis']['primary\_field']

skills = resume\_data['skills']

# Search by field + skills combinations

job\_results = []

# Primary field search

jobs\_1 = self.job\_searcher.search\_jobs(

keywords=primary\_field,

location=search\_params['location'],

salary\_min=search\_params['min\_salary']

)

# Skills-based searches

for skill in skills[:3]: # Top 3 skills

jobs\_skill = self.job\_searcher.search\_jobs(

keywords=f"{skill} {primary\_field}",

location=search\_params['location'],

salary\_min=search\_params['min\_salary']

)

job\_results.extend(jobs\_skill)

# Remove duplicates and return

return self.\_deduplicate\_jobs(job\_results)

**Step 3: Integration Workflow**

class MingusJobRecommendationEngine:

def \_\_init\_\_(self):

self.resume\_parser = AdvancedResumeParser()

self.income\_comparator = IncomeComparator()

self.job\_matcher = IncomeBoostJobMatcher()

def generate\_recommendations(self, resume\_file, demographic\_data):

"""Complete workflow from resume to targeted job recommendations"""

# Step 1: Enhanced resume analysis

resume\_data = self.resume\_parser.parse\_resume(resume\_file)

field\_analysis = self.resume\_parser.analyze\_field\_expertise(resume\_data)

resume\_data['field\_analysis'] = field\_analysis

# Step 2: Income context analysis

income\_comparisons = self.income\_comparator.get\_comprehensive\_comparison(

user\_income=demographic\_data['current\_salary'],

age=demographic\_data['age'],

race=demographic\_data['race'],

education=demographic\_data['education'],

location=demographic\_data['location']

)

income\_context = self.\_build\_income\_context(income\_comparisons, demographic\_data)

# Step 3: Targeted job search and selection

job\_recommendations = self.job\_matcher.find\_targeted\_opportunities(

resume\_data, demographic\_data, income\_context

)

# Step 4: Build complete recommendation package

complete\_analysis = {

'resume\_analysis': resume\_data,

'income\_context': income\_context,

'job\_recommendations': job\_recommendations,

'action\_plan': self.\_generate\_action\_plan(job\_recommendations),

'financial\_impact': self.\_calculate\_financial\_impact(income\_context, job\_recommendations)

}

return complete\_analysis

**Implementation Timeline**

**Week 1: Core Enhancement**

* **Day 1-2**: Enhanced resume parser with field detection
* **Day 3-4**: Income-aware job matching algorithm
* **Day 5**: Integration and testing

**Week 2: Optimization & Polish**

* **Day 1-2**: Three-tier recommendation selection
* **Day 3-4**: User interface and results display
* **Day 5**: Performance optimization and deployment

**Expected Output Format**

**User-Facing Recommendation**

🎯 YOUR INCOME ADVANCEMENT OPPORTUNITIES

Based on your resume analysis and income comparison:

• You're currently in the 35th percentile for Data Analysts in Atlanta

• Target income range: $65,000 - $78,000 (18-42% increase)

• Your skills align well with growth opportunities in your field

📊 THREE TARGETED OPPORTUNITIES:

1. CONSERVATIVE GROWTH (92% match)

Senior Business Analyst at Delta Air Lines

💰 $65,000 - $68,000 (18-24% increase)

✅ Strong skill alignment - apply immediately

📈 Moves you to 58th percentile in your field

2. OPTIMAL GROWTH (88% match)

Data Analyst II at Chick-fil-A Corporate

💰 $68,000 - $75,000 (24-36% increase)

📚 Complete Tableau basics course first

📈 Moves you to 72nd percentile in your field

3. STRETCH OPPORTUNITY (78% match)

Product Analyst at Mailchimp (Intuit)

💰 $75,000 - $85,000 (36-55% increase)

🚀 Network + product analytics course

📈 Moves you to 85th percentile in your field

💡 NEXT STEPS:

1. Apply to Delta position this week

2. Start Tableau course on Coursera

3. Connect with Mailchimp employees on LinkedIn

**Success Metrics**

* **Precision**: 3 highly relevant jobs vs. 20 generic results
* **Income Focus**: All recommendations offer 15%+ salary increases
* **Actionability**: Clear next steps for each opportunity
* **Personalization**: Specific to user's field, location, and income goals
* **Motivation**: Shows clear path from current position to financial advancement

This process transforms generic job searching into targeted career advancement planning, directly addressing the income gaps identified in your target demographic.

**Cursor Prompts for Targeted Job Recommendation Engine**

**Context for All Prompts**

**Project**: Mingus targeted job recommendation engine for African American professionals aged 25-35 earning $40k-100k **Goal**: Transform resume analysis into 3 specific, income-boosting job recommendations in user's MSA and field**Strategy**: Conservative (15-20% increase), Optimal (25-30% increase), Stretch (35%+ increase) opportunities**Integration**: Builds on existing resume parser and income comparison features

**Prompt 1: Enhanced Resume Parser with Field Detection**

Create an advanced resume parser that extends the existing ResumeParser class with field expertise analysis and career trajectory detection.

Context: This enhances Mingus resume analysis to identify not just skills, but the user's primary field of expertise, experience level, and career growth potential. This enables targeted job recommendations that match both their background and income advancement goals.

Requirements:

1. Extend existing ResumeParser class with field detection capabilities

2. Classify users into primary fields: Data Analysis, Project Management, Software Development, Marketing, Finance, Sales, Operations, HR

3. Determine experience level: Entry (0-2 years), Mid (3-5 years), Senior (5+ years)

4. Identify career trajectory and growth potential

5. Extract leadership indicators and advancement readiness

6. Calculate technical vs business skills ratio

7. Detect industry focus and transferable skills

Key features needed:

- Field classification algorithm using keyword analysis and job title patterns

- Experience level calculation based on job titles, responsibilities, and tenure

- Career progression analysis to predict next logical career steps

- Leadership potential scoring based on resume language

- Skills categorization (technical, business, soft skills) with proficiency indicators

- Industry experience detection for targeted job searching

Field classification criteria:

- Data Analysis: SQL, Python, Tableau, Analytics, Statistics, Excel, Reporting

- Project Management: PMP, Agile, Scrum, Stakeholder Management, Timeline Management

- Software Development: Programming languages, Frameworks, APIs, Git, DevOps

- Marketing: Campaigns, Social Media, SEO, Google Analytics, Content Creation

- Finance: Financial Modeling, Accounting, Budgeting, Forecasting, Analysis

- Sales: CRM, Salesforce, Lead Generation, Client Relations, Revenue Growth

Experience level indicators:

- Entry: Intern, Junior, Assistant, Associate, Coordinator roles

- Mid: Analyst, Specialist, Senior, Lead roles with 3-5 years experience

- Senior: Manager, Director, Principal, Head roles with leadership responsibilities

The enhanced parser should return structured data including:

- Primary and secondary fields of expertise

- Experience level with confidence score

- Career trajectory analysis

- Skills categorization and proficiency levels

- Leadership readiness indicators

- Industry experience summary

Include comprehensive error handling and logging for debugging field classification accuracy.

**Prompt 2: Income-Aware Job Matching Algorithm**

Create an intelligent job matching system that prioritizes income advancement opportunities based on resume analysis and demographic income comparisons.

Context: This is the core of Mingus targeted job recommendations. Instead of generic job matching, this system specifically finds opportunities that offer 15-45% salary increases while matching the user's field expertise and location preferences.

Requirements:

1. Integration with existing JobSearcher and IncomeComparator classes

2. Salary-focused search parameters based on user's income gap analysis

3. Multi-dimensional job scoring prioritizing income improvement

4. Field-specific job search strategies using enhanced resume analysis

5. MSA-targeted search for Atlanta, Houston, DC, Dallas, NYC, Philadelphia, Chicago, Charlotte, Miami, Baltimore

6. Company quality assessment for reliable salary information

7. Remote work opportunity identification for expanded income potential

Key algorithms needed:

- Income-based search parameter calculation (minimum 15% increase requirement)

- Comprehensive job scoring system weighing salary improvement (35%), skills match (25%), career progression (20%), company quality (10%), location fit (5%), industry alignment (5%)

- Multi-source job aggregation with deduplication across APIs

- Salary range validation and normalization across different job posting formats

- Company tier classification (Fortune 500, Growth companies, Startups) for compensation reliability

Search strategy features:

- Primary field keyword optimization (e.g., "Data Analyst Python" vs generic "Data")

- Skills-based secondary searches for expanded opportunities

- Location flexibility including remote opportunities with local companies

- Salary filtering with buffer zones to catch negotiable positions

- Industry crossover identification for career pivots with income growth

Job scoring components:

- Salary improvement score: Percentage increase vs current income with bonus for reaching target percentiles

- Skills alignment score: Percentage match of required vs possessed skills

- Career progression score: Logical next step vs lateral move assessment

- Company stability score: Based on company size, funding, and Glassdoor data

- Location compatibility score: Commute time, remote options, relocation requirements

- Growth potential score: Industry trends and company expansion indicators

The system should filter out jobs below 15% salary increase threshold and prioritize opportunities offering 25%+ increases while maintaining realistic skill alignment requirements.

Include comprehensive logging for debugging search effectiveness and recommendation quality.

**Prompt 3: Three-Tier Job Selection Engine**

Create a sophisticated job selection algorithm that chooses exactly 3 opportunities representing Conservative, Optimal, and Stretch career advancement options.

Context: Mingus users need specific, actionable recommendations rather than overwhelming job lists. This system selects one job from each risk/reward tier to provide a complete career advancement strategy.

Requirements:

1. Three-tier selection strategy ensuring diverse opportunity levels

2. Conservative tier: 15-20% income increase, 80%+ skills match, low risk

3. Optimal tier: 25-30% income increase, 70%+ skills match, medium risk

4. Stretch tier: 35%+ income increase, 60%+ skills match, higher risk/reward

5. Skill gap analysis and upskilling recommendations for each opportunity

6. Application strategy guidance specific to each opportunity level

7. Income percentile impact calculation for each recommendation

Selection algorithm features:

- Tier classification based on salary increase percentage and skill requirements

- Diversity enforcement ensuring geographic and company variety across tiers

- Risk assessment considering skill gaps, company stability, and role demands

- Opportunity scoring within each tier to select the best representative

- Backup selection in case primary choices become unavailable

Recommendation package for each job:

- Job details (title, company, salary range, location)

- Income impact analysis (percentage increase, new percentile ranking)

- Skills match breakdown (possessed vs required vs nice-to-have)

- Missing skills identification with learning recommendations

- Application strategy (immediate apply, upskill first, networking required)

- Success probability assessment based on profile alignment

- Timeline to application readiness

Conservative tier criteria:

- 15-20% salary increase minimum

- 80%+ skills match with minimal gaps

- Established companies with clear hiring processes

- Roles requiring skills user already possesses

- High probability of interview if applied

Optimal tier criteria:

- 25-30% salary increase target

- 70%+ skills match with learnable gaps

- Growth companies or established firms expanding teams

- Roles requiring 1-2 additional skills user can acquire

- Medium effort required for application success

Stretch tier criteria:

- 35%+ salary increase potential

- 60%+ skills match with significant but achievable gaps

- High-growth companies or premium employers

- Roles representing clear career advancement

- Requires strategic preparation and networking

Include detailed reasoning for each selection and alternative options in case primary recommendations are not suitable.

**Prompt 4: Complete Recommendation Engine Integration**

Create the master MingusJobRecommendationEngine class that orchestrates the complete workflow from resume upload to targeted job recommendations.

Context: This is the main controller that integrates enhanced resume parsing, income comparison, and targeted job selection into a seamless user experience for Mingus career advancement feature.

Requirements:

1. Integration of AdvancedResumeParser, IncomeComparator, and IncomeBoostJobMatcher

2. Complete workflow orchestration from file upload to final recommendations

3. Error handling and graceful degradation for each processing step

4. Performance optimization for sub-3-second total processing time

5. Data validation and sanitization throughout the pipeline

6. Comprehensive logging for debugging and analytics

7. Caching strategies for repeat users and common searches

Main workflow components:

- File validation and resume processing with enhanced field detection

- Demographic data integration with income comparison analysis

- Targeted job search execution with salary-focused filtering

- Three-tier job selection with detailed recommendation packages

- Financial impact calculation showing income improvement potential

- Action plan generation with specific next steps for each opportunity

Integration features:

- Unified error handling across all components

- Progress tracking for user feedback during processing

- Result caching for performance improvement

- Data consistency validation between components

- Fallback strategies when APIs or services fail

Output format requirements:

- Complete user profile analysis including field expertise and income position

- Three targeted job recommendations with detailed justification

- Financial impact analysis showing percentile improvements

- Specific action plan with timelines and skill development recommendations

- Success probability assessments for each opportunity

Performance targets:

- Resume processing: <2 seconds

- Income comparison: <1 second

- Job search and selection: <5 seconds

- Total workflow: <8 seconds end-to-end

Error handling scenarios:

- Resume parsing failures with user-friendly error messages

- Income comparison data unavailable with fallback analysis

- Job search API failures with alternative search strategies

- Insufficient job results with expanded search criteria

The engine should provide comprehensive analytics data for tracking feature effectiveness and user success rates.

Include extensive documentation and example usage demonstrating the complete user journey from resume upload to receiving targeted career advancement recommendations.

**Prompt 5: Flask Integration and API Endpoints**

Integrate the MingusJobRecommendationEngine with the existing Flask application and create new API endpoints for the targeted job recommendation feature.

Context: Enhance the existing Mingus Flask app to support the new targeted job recommendation workflow while maintaining compatibility with existing resume parsing and income comparison features.

Requirements:

1. Update existing upload route to support enhanced demographic collection

2. Create new processing endpoint for targeted job recommendations

3. Add API endpoint for mobile/external integration capabilities

4. Implement progress tracking for long-running job searches

5. Add result caching and session management for user experience

6. Create admin endpoints for monitoring recommendation quality

7. Integrate with existing template system for seamless user experience

New route implementations:

- Enhanced /upload route with comprehensive demographic form

- /process-recommendations route for job recommendation generation

- /api/recommendations endpoint for programmatic access

- /progress endpoint for real-time processing updates

- /admin/analytics endpoint for recommendation effectiveness tracking

Form enhancements needed:

- Additional demographic fields for precise income comparison

- Career goals and salary expectations collection

- Preferred job search parameters (remote work, company size, industry)

- Skills assessment and learning preferences

- Geographic flexibility and relocation willingness

Processing workflow:

- Asynchronous job processing for better user experience

- Progress updates during resume analysis, income comparison, and job search

- Real-time status communication to frontend

- Error recovery and alternative processing paths

- Result persistence for user access and analytics

API design features:

- RESTful endpoints with proper HTTP status codes

- JSON response format with comprehensive recommendation data

- Rate limiting to prevent abuse while allowing legitimate usage

- Authentication consideration for future premium features

- Response caching for improved performance

Session management:

- User session tracking for multi-step recommendation process

- Temporary storage of uploaded files and processing results

- Progress state persistence across page refreshes

- Result history for returning users

Security considerations:

- File upload validation and virus scanning

- Input sanitization for all demographic data

- Rate limiting on processing endpoints

- Secure handling of salary and personal information

- CSRF protection for form submissions

Integration with existing templates:

- Enhanced results.html with three-tier job recommendations

- Progress indicator templates for processing feedback

- Error handling templates for graceful failure communication

- Mobile-responsive design for target demographic usage patterns

Include comprehensive error handling, logging, and monitoring capabilities for production deployment.

**Prompt 6: Enhanced Results Interface and User Experience**

Create a comprehensive user interface for displaying targeted job recommendations with clear visual hierarchy and actionable next steps.

Context: Transform the job recommendation results into a compelling, easy-to-understand interface that motivates users to take action on their career advancement opportunities.

Requirements:

1. Three-tier recommendation display with clear visual differentiation

2. Income impact visualization showing percentile improvements

3. Skills gap analysis with learning recommendations

4. Application strategy guidance for each opportunity

5. Progress tracking for recommendation follow-up

6. Mobile-first responsive design for target demographic

7. Accessibility compliance for inclusive user experience

Visual design components:

- Hero section summarizing user's income position and opportunity potential

- Three distinct cards for Conservative, Optimal, and Stretch recommendations

- Income improvement visualizations (percentage increases, percentile charts)

- Skills alignment indicators with gap identification

- Company information and credibility indicators

- Application readiness assessment and timeline guidance

Recommendation card features:

- Job title, company, and salary range prominence

- Match score and risk level indication

- Income impact calculation with specific dollar amounts

- Skills breakdown (possessed, required, missing)

- Application strategy with specific next steps

- Success probability and timeline estimates

- Direct links to job postings and company research

Interactive elements:

- Expandable job descriptions and company details

- Skills gap learning resource recommendations

- Application tracking and reminder system

- Salary negotiation guidance and market data

- Networking suggestions for each opportunity

User engagement features:

- Save recommendations for later review

- Track application progress and outcomes

- Request updated recommendations based on skill development

- Share success stories and referral opportunities

- Calendar integration for application deadlines and interview scheduling

Mobile optimization:

- Touch-friendly interface elements

- Condensed information hierarchy for small screens

- Swipe navigation between recommendation tiers

- Quick action buttons for immediate next steps

- Offline access to saved recommendations

Motivational messaging:

- Positive framing of income improvement opportunities

- Clear connection between current position and advancement potential

- Specific dollar amounts and percentile improvements

- Success story integration and peer comparisons

- Progress celebration and milestone recognition

Accessibility features:

- Screen reader compatibility for all content

- High contrast mode for financial data readability

- Keyboard navigation support throughout interface

- Alternative text for data visualizations and charts

- Voice interface consideration for hands-free usage

The interface should feel like a professional career coaching session with specific, actionable guidance rather than a generic job board experience.

Include CSS animations and micro-interactions that suggest upward career movement and financial progress.

**Prompt 7: Testing and Quality Assurance Suite**

Create comprehensive testing suite for the targeted job recommendation engine including unit tests, integration tests, and user scenario validation.

Context: The job recommendation engine affects users' career decisions and financial futures, requiring extensive testing to ensure accuracy, reliability, and appropriate recommendations for the target demographic.

Requirements:

1. Unit tests for all recommendation engine components

2. Integration tests for complete workflow scenarios

3. Performance testing for processing time requirements

4. Recommendation quality validation with sample user profiles

5. Edge case testing for various resume and demographic combinations

6. Mock data generation for consistent testing environments

7. User acceptance testing scenarios for target demographic

Component testing coverage:

- AdvancedResumeParser field detection accuracy

- IncomeBoostJobMatcher salary filtering and scoring algorithms

- Three-tier selection engine diversity and appropriateness

- MingusJobRecommendationEngine workflow orchestration

- Flask integration endpoint functionality

- Results interface rendering and user interaction

Test data requirements:

- Sample resumes representing target demographic profiles

- Mock income comparison data for various geographic and demographic scenarios

- Simulated job search API responses with realistic salary and requirement data

- Edge case resumes (sparse information, unusual career paths, multiple fields)

- Performance benchmarking data for processing time validation

Quality assurance scenarios:

- African American professionals across target age range (25-35)

- Various education levels and career stages

- Different metropolitan areas with varying cost of living

- Career changers and industry switchers

- Underemployed professionals seeking advancement

- High performers ready for stretch opportunities

Recommendation validation criteria:

- All recommendations offer minimum 15% salary increase

- Skills alignment accuracy within acceptable thresholds

- Geographic targeting appropriate for user location

- Company quality and legitimacy verification

- Application strategy appropriateness for opportunity level

- Income percentile calculations accuracy

Performance testing targets:

- Resume processing under 2 seconds

- Income comparison under 1 second

- Job search and selection under 5 seconds

- Total workflow under 8 seconds end-to-end

- Concurrent user handling up to 50 simultaneous requests

- Memory usage optimization for hosting cost management

Error scenario testing:

- Corrupted or unreadable resume files

- Incomplete demographic information

- Job search API failures and timeouts

- Insufficient job results for user criteria

- Network connectivity issues during processing

- Database failures and recovery procedures

User experience testing:

- Interface usability across devices and browsers

- Accessibility compliance for users with disabilities

- Loading time perception and progress communication

- Error message clarity and recovery guidance

- Mobile responsiveness for primary usage patterns

Include automated testing pipeline integration and continuous monitoring for production recommendation quality and system performance.

**Prompt 8: Analytics and Performance Optimization**

Create comprehensive analytics tracking and performance optimization system for the targeted job recommendation engine.

Context: Track user engagement, recommendation effectiveness, and system performance to continuously improve the Mingus career advancement feature and measure business impact.

Requirements:

1. User behavior analytics throughout recommendation workflow

2. Recommendation quality metrics and success rate tracking

3. Performance monitoring and optimization for production scale

4. A/B testing framework for recommendation algorithm improvements

5. Business analytics for feature ROI and user satisfaction

6. Cost optimization for API usage and hosting resources

7. Real-time monitoring and alerting for system health

Analytics implementation:

- User journey tracking from resume upload to application actions

- Recommendation click-through rates and engagement metrics

- Application success rates and salary increase achievements

- User retention and feature usage patterns

- Demographic correlation analysis for recommendation effectiveness

- Geographic performance differences across target metro areas

Performance optimization features:

- Response time monitoring and optimization alerts

- Database query optimization for recommendation retrieval

- Caching strategies for frequently accessed recommendation patterns

- API usage optimization to stay within free tier limits

- Memory usage profiling and garbage collection tuning

- CDN integration for static asset delivery

Business metrics tracking:

- Feature conversion rates from upload to recommendations

- User satisfaction scores and feedback analysis

- Premium tier conversion potential based on feature usage

- Word-of-mouth referral tracking and attribution

- Cost per successful job placement or salary increase

- Revenue impact from improved user financial outcomes

A/B testing framework:

- Recommendation algorithm variant testing

- User interface layout and messaging optimization

- Demographic targeting refinement experiments

- Income targeting threshold optimization

- Skills matching algorithm improvements

- Application strategy effectiveness testing

Monitoring and alerting:

- Real-time system health monitoring

- Recommendation quality degradation alerts

- API failure detection and recovery procedures

- User error rate monitoring and investigation

- Performance threshold breaches and escalation

- Security monitoring for unusual usage patterns

Cost optimization strategies:

- API usage monitoring and optimization recommendations

- Hosting resource scaling based on demand patterns

- Database optimization for query efficiency

- CDN usage optimization for global performance

- Background job processing optimization

- Resource usage forecasting for budget planning

Data privacy and compliance:

- User data anonymization for analytics purposes

- GDPR compliance for user tracking and data retention

- Opt-out mechanisms for analytics participation

- Data export capabilities for user data portability

- Audit logging for compliance and debugging purposes

The analytics system should provide actionable insights for improving recommendation quality, user satisfaction, and business outcomes while maintaining user privacy and system performance.

Include dashboard creation for real-time monitoring and reporting capabilities for business stakeholders.

**Implementation Sequence**

**Week 1: Core Engine Development**

* **Day 1**: Prompt 1 (Enhanced Resume Parser)
* **Day 2**: Prompt 2 (Income-Aware Job Matching)
* **Day 3**: Prompt 3 (Three-Tier Selection)
* **Day 4**: Prompt 4 (Complete Integration)
* **Day 5**: Testing and refinement

**Week 2: User Experience and Production**

* **Day 1**: Prompt 5 (Flask Integration)
* **Day 2**: Prompt 6 (Enhanced UI)
* **Day 3**: Prompt 7 (Testing Suite)
* **Day 4**: Prompt 8 (Analytics & Optimization)
* **Day 5**: Production deployment and monitoring

**Cursor Usage Tips**

**Sequential Development:**

# Use these commands to reference previous work

@advanced\_resume\_parser.py @income\_boost\_job\_matcher.py Ensure data structures are compatible

# Cross-reference for integration

@app.py Add the MingusJobRecommendationEngine to existing workflow

# Optimize performance

Optimize this job matching algorithm for processing under 5 seconds

# Test edge cases

Add comprehensive error handling for when job search APIs return no results

**Validation Commands:**

# Test with sample data

Create test scenarios for African American professionals in Atlanta with $45k-65k salaries

# Verify calculations

Validate that income percentile calculations match expected demographic data

# Check user experience

Ensure the three-tier recommendation display is mobile-responsive and accessible

Each prompt builds a complete production-ready system that transforms resume analysis into targeted career advancement recommendations specifically designed for your Mingus demographic.