

# PsychSync - TaskMaster Project Plan

## Project Overview

**Project:** PsychSync - Team Psychology & Optimization Platform

**Duration:** 16-20 weeks

**Team Size:** 6-8 developers

**Budget:** \$800K - \$1.2M

**Launch Target:** Q3 2024

## Phase 1: Foundation & Planning (Weeks 1-2)

### Sprint 1.1: Project Setup & Architecture (Week 1)

**Objective:** Establish project foundation and technical architecture

#### Epic: Project Foundation

☐ **Repository Setup**

- Create GitHub organization and repositories
- Set up branch protection rules and CI/CD pipelines
- Configure development environments
- **Assignee:** DevOps Lead | **Effort:** 8h | **Priority:** High

☐ **Tech Stack Finalization**

- Finalize technology decisions and versions
- Set up development toolchain
- Create development environment documentation
- **Assignee:** Tech Lead | **Effort:** 12h | **Priority:** High

☐ **Database Design**

- Create detailed ERD for all entities
- Set up database schemas and migrations
- Configure database environments
- **Assignee:** Backend Lead | **Effort:** 16h | **Priority:** High

#### Epic: Core Infrastructure

☐ **API Framework Setup**

- Initialize FastAPI project structure
- Configure authentication middleware
- Set up OpenAPI documentation

- **Assignee:** Backend Developer 1 | **Effort:** 12h | **Priority:** High

#### ☐ **Frontend Foundation**

- Initialize React application with TypeScript
- Set up routing and state management
- Configure build and development tools
- **Assignee:** Frontend Lead | **Effort:** 12h | **Priority:** High

## **Sprint 1.2: Core Authentication & User Management (Week 2)**

**Objective:** Implement user authentication and basic user management

### **Epic: Authentication System**

#### ☐ **JWT Authentication**

- Implement JWT token generation and validation
- Create login/logout endpoints
- Set up refresh token mechanism
- **Assignee:** Backend Developer 1 | **Effort:** 16h | **Priority:** High

#### ☐ **User Registration**

- Create user registration endpoint
- Implement email verification
- Set up password reset functionality
- **Assignee:** Backend Developer 2 | **Effort:** 12h | **Priority:** High

#### ☐ **Frontend Auth Components**

- Create Login and Register components
- Implement protected route wrapper
- Set up authentication state management
- **Assignee:** Frontend Developer 1 | **Effort:** 14h | **Priority:** High

## **Phase 2: Core Features Development (Weeks 3-8)**

### **Sprint 2.1: Team Management Foundation (Week 3)**

**Objective:** Build basic team management functionality

### **Epic: Team CRUD Operations**

#### ☐ **Team Entity Management**

- Create team creation and update endpoints
- Implement team deletion with safeguards
- Set up team listing and filtering

- **Assignee:** Backend Developer 2 | **Effort:** 16h | **Priority:** High

#### ☐ **Team Membership**

- Implement add/remove team members
- Create role-based permissions
- Set up team hierarchy system
- **Assignee:** Backend Developer 1 | **Effort:** 18h | **Priority:** High

#### ☐ **Team Management UI**

- Create team listing and detail views
- Implement team creation wizard
- Build member management interface
- **Assignee:** Frontend Developer 2 | **Effort:** 20h | **Priority:** High

## **Sprint 2.2: Assessment Framework Core (Week 4)**

**Objective:** Implement assessment framework and question management

### **Epic: Assessment Infrastructure**

#### ☐ **Framework Management**

- Create assessment framework models
- Implement framework CRUD operations
- Set up question bank management
- **Assignee:** Backend Developer 3 | **Effort:** 18h | **Priority:** High

#### ☐ **Question Engine**

- Build question delivery system
- Implement response collection
- Create scoring algorithms
- **Assignee:** Backend Developer 1 | **Effort:** 20h | **Priority:** High

#### ☐ **Assessment UI Components**

- Create assessment taking interface
- Build question rendering components
- Implement progress tracking
- **Assignee:** Frontend Developer 1 | **Effort:** 22h | **Priority:** High

## **Sprint 2.3: Assessment Results & Analysis (Week 5)**

**Objective:** Build assessment scoring and result presentation

### **Epic: Results Processing**

### ☐ **Scoring Engine**

- Implement MBTI scoring algorithm
- Create Big Five personality scoring
- Set up DISC assessment scoring
- **Assignee:** Data Scientist | **Effort:** 24h | **Priority:** High

### ☐ **Results Storage**

- Create assessment results endpoints
- Implement result retrieval and filtering
- Set up result comparison features
- **Assignee:** Backend Developer 2 | **Effort:** 16h | **Priority:** High

### ☐ **Results Visualization**

- Create personality profile charts
- Build assessment comparison views
- Implement result export functionality
- **Assignee:** Frontend Developer 3 | **Effort:** 20h | **Priority:** High

## **Sprint 2.4: Basic Analytics Dashboard (Week 6)**

**Objective:** Create foundational analytics and dashboard

### **Epic: Dashboard Infrastructure**

#### ☐ **Metrics Calculation**

- Implement team compatibility scoring
- Create performance metrics aggregation
- Set up trend analysis calculations
- **Assignee:** Data Scientist | **Effort:** 20h | **Priority:** High

#### ☐ **Dashboard API**

- Create dashboard data endpoints
- Implement real-time metrics updates
- Set up data caching for performance
- **Assignee:** Backend Developer 3 | **Effort:** 14h | **Priority:** High

#### ☐ **Dashboard UI**

- Build main dashboard layout
- Create metric cards and charts
- Implement responsive dashboard design
- **Assignee:** Frontend Lead | **Effort:** 18h | **Priority:** High

## Sprint 2.5: Team Optimization Engine (Week 7)

**Objective:** Implement AI-powered team optimization

### Epic: Optimization Algorithms

#### ☐ Compatibility Analysis

- Build personality compatibility algorithms
- Implement team balance scoring
- Create conflict prediction models
- **Assignee:** Data Scientist | **Effort:** 28h | **Priority:** High

#### ☐ Recommendation Engine

- Create team composition recommendations
- Implement role assignment suggestions
- Build optimization report generation
- **Assignee:** ML Engineer | **Effort:** 24h | **Priority:** High

#### ☐ Optimization UI

- Build team optimization interface
- Create recommendation display components
- Implement optimization controls
- **Assignee:** Frontend Developer 2 | **Effort:** 20h | **Priority:** High

## Sprint 2.6: Advanced Analytics (Week 8)

**Objective:** Build comprehensive analytics and reporting

### Epic: Advanced Analytics

#### ☐ Behavioral Trend Analysis

- Implement trend detection algorithms
- Create pattern recognition for team dynamics
- Set up predictive analytics models
- **Assignee:** Data Scientist | **Effort:** 26h | **Priority:** Medium

#### ☐ Performance Metrics

- Build velocity prediction models
- Create productivity correlation analysis
- Implement team health scoring
- **Assignee:** ML Engineer | **Effort:** 22h | **Priority:** Medium

#### ☐ Analytics Visualization

- Create advanced chart components

- Build interactive analytics dashboard
- Implement data export features
- **Assignee:** Frontend Developer 3 | **Effort:** 24h | **Priority:** Medium

## Phase 3: Enhancement & Integration (Weeks 9-12)

### Sprint 3.1: Advanced Features (Week 9)

**Objective:** Implement advanced platform features

#### Epic: Advanced Functionality

##### ☐ Multi-Framework Support

- Integrate additional assessment frameworks
- Implement cross-framework comparisons
- Create framework recommendation engine
- **Assignee:** Backend Developer 1 | **Effort:** 20h | **Priority:** Medium

##### ☐ Advanced Team Features

- Implement team templates
- Create team cloning functionality
- Build team performance tracking
- **Assignee:** Backend Developer 2 | **Effort:** 18h | **Priority:** Medium

##### ☐ Enhanced User Experience

- Implement drag-and-drop team builder
- Create guided onboarding flow
- Build interactive tutorials
- **Assignee:** Frontend Developer 1 | **Effort:** 22h | **Priority:** Medium

### Sprint 3.2: Integration & Third-Party Services (Week 10)

**Objective:** Integrate with external services and platforms

#### Epic: External Integrations

##### ☐ Slack Integration

- Build Slack bot for team insights
- Implement notification delivery to Slack
- Create Slack slash commands
- **Assignee:** Backend Developer 3 | **Effort:** 20h | **Priority:** Medium

##### ☐ Microsoft Teams Integration

- Create Teams app integration

- Implement SSO with Microsoft
- Build Teams notification system
- **Assignee:** Backend Developer 1 | **Effort:** 18h | **Priority:** Medium

#### ☐ **Email Service Integration**

- Set up transactional email service
- Create email templates
- Implement email scheduling
- **Assignee:** Backend Developer 2 | **Effort:** 12h | **Priority:** Medium

#### ☐ **Calendar Integration**

- Integrate with Google Calendar
- Implement meeting scheduling
- Create assessment reminders
- **Assignee:** Frontend Developer 2 | **Effort:** 16h | **Priority:** Low

### **Sprint 3.3: Mobile Responsiveness & PWA (Week 11)**

**Objective:** Ensure mobile compatibility and progressive web app features

#### **Epic: Mobile Experience**

##### ☐ **Responsive Design**

- Optimize all components for mobile
- Implement mobile-first navigation
- Create touch-friendly interfaces
- **Assignee:** Frontend Developer 3 | **Effort:** 24h | **Priority:** High

##### ☐ **Progressive Web App**

- Implement service worker
- Add offline functionality
- Create app manifest
- **Assignee:** Frontend Lead | **Effort:** 16h | **Priority:** Medium

##### ☐ **Mobile Assessment Experience**

- Optimize assessment taking on mobile
- Implement swipe gestures
- Create mobile-specific components
- **Assignee:** Frontend Developer 1 | **Effort:** 18h | **Priority:** High

### **Sprint 3.4: Advanced Security & Compliance (Week 12)**

**Objective:** Implement enterprise-grade security and compliance features

## Epic: Security Enhancement

### ☐ GDPR Compliance

- Implement data export functionality
- Create data deletion workflows
- Add consent management
- **Assignee:** Backend Lead | **Effort:** 20h | **Priority:** High

### ☐ SOC 2 Compliance

- Implement audit logging
- Create access control reports
- Set up security monitoring
- **Assignee:** DevOps Lead | **Effort:** 18h | **Priority:** High

### ☐ Enterprise SSO

- Implement SAML 2.0 support
- Add LDAP integration
- Create SSO configuration UI
- **Assignee:** Backend Developer 3 | **Effort:** 22h | **Priority:** Medium

## Phase 4: Testing & Quality Assurance (Weeks 13-16)

### Sprint 4.1: Automated Testing (Week 13)

**Objective:** Implement comprehensive automated testing

#### Epic: Test Coverage

### ☐ Backend Testing

- Unit tests for all services (90%+ coverage)
- Integration tests for API endpoints
- Database migration tests
- **Assignee:** Backend Team | **Effort:** 32h | **Priority:** High

### ☐ Frontend Testing

- Component unit tests
- Integration tests for user flows
- E2E tests with Cypress
- **Assignee:** Frontend Team | **Effort:** 28h | **Priority:** High

### ☐ Performance Testing

- Load testing with k6
- Stress testing for API endpoints



- Database performance testing
- **Assignee:** QA Lead | **Effort:** 20h | **Priority:** High

## **Sprint 4.2: Manual Testing & QA (Week 14)**

**Objective:** Conduct thorough manual testing and quality assurance

### **Epic: Quality Assurance**

#### ☐ **Functional Testing**

- Test all user workflows
- Verify assessment accuracy
- Validate team optimization algorithms
- **Assignee:** QA Team | **Effort:** 40h | **Priority:** High

#### ☐ **Usability Testing**

- Conduct user acceptance testing
- Gather feedback from beta users
- Perform accessibility audits
- **Assignee:** UX Designer + QA | **Effort:** 24h | **Priority:** High

#### ☐ **Security Testing**

- Penetration testing
- Vulnerability assessments
- Security code review
- **Assignee:** Security Consultant | **Effort:** 16h | **Priority:** High

## **Sprint 4.3: Performance Optimization (Week 15)**

**Objective:** Optimize system performance and resolve bottlenecks

### **Epic: Performance Tuning**

#### ☐ **Database Optimization**

- Query optimization
- Index tuning
- Connection pool optimization
- **Assignee:** Database Administrator | **Effort:** 20h | **Priority:** High

#### ☐ **API Performance**

- Endpoint response time optimization
- Caching implementation
- Rate limiting configuration

- **Assignee:** Backend Lead | **Effort:** 16h | **Priority:** High

#### ☐ **Frontend Optimization**

- Bundle size optimization
- Lazy loading implementation
- Performance monitoring setup
- **Assignee:** Frontend Lead | **Effort:** 18h | **Priority:** High

### **Sprint 4.4: Bug Fixes & Polish (Week 16)**

**Objective:** Address bugs and polish user experience

#### **Epic: Quality Polish**

##### ☐ **Bug Resolution**

- Fix all critical and high-priority bugs
- Address usability issues
- Implement user feedback
- **Assignee:** All Developers | **Effort:** 32h | **Priority:** High

##### ☐ **UI/UX Polish**

- Refine user interface elements
- Improve user experience flows
- Add micro-interactions and animations
- **Assignee:** Frontend Team | **Effort:** 24h | **Priority:** Medium

### **Phase 5: Deployment & Launch (Weeks 17-20)**

#### **Sprint 5.1: Production Deployment (Week 17)**

**Objective:** Deploy application to production environment

#### **Epic: Production Setup**

##### ☐ **Infrastructure Deployment**

- Set up production Kubernetes cluster
- Configure load balancers and CDN
- Implement monitoring and logging
- **Assignee:** DevOps Lead | **Effort:** 24h | **Priority:** High

##### ☐ **Database Migration**

- Set up production database cluster
- Implement backup and recovery procedures
- Configure monitoring and alerts

- **Assignee:** Database Administrator | **Effort:** 16h | **Priority:** High

#### ☐ **Security Configuration**

- Configure SSL certificates
- Set up WAF and security headers
- Implement secrets management
- **Assignee:** DevOps Lead | **Effort:** 12h | **Priority:** High

### **Sprint 5.2: Beta Testing & Feedback (Week 18)**

**Objective:** Conduct beta testing with select customers

#### **Epic: Beta Program**

#### ☐ **Beta User Onboarding**

- Set up beta user accounts
- Create onboarding documentation
- Implement feedback collection system
- **Assignee:** Product Manager | **Effort:** 16h | **Priority:** High

#### ☐ **Monitoring & Support**

- Set up real-time monitoring
- Create support ticket system
- Implement user analytics
- **Assignee:** DevOps + Support | **Effort:** 20h | **Priority:** High

### **Sprint 5.3: Launch Preparation (Week 19)**

**Objective:** Prepare for public launch

#### **Epic: Launch Readiness**

#### ☐ **Documentation Completion**

- Finalize user documentation
- Create API documentation
- Write deployment guides
- **Assignee:** Technical Writer | **Effort:** 24h | **Priority:** High

#### ☐ **Marketing Materials**

- Create product demo videos
- Build landing page
- Prepare press release materials
- **Assignee:** Marketing Team | **Effort:** 32h | **Priority:** Medium

## ☐ Final Testing

- Conduct final regression testing
- Perform disaster recovery testing
- Validate all integrations
- **Assignee:** QA Team | **Effort:** 28h | **Priority:** High

## Sprint 5.4: Launch & Post-Launch Support (Week 20)

**Objective:** Execute launch and provide immediate support

### Epic: Launch Execution

## ☐ Go-Live Activities

- Execute production deployment
- Monitor system performance
- Provide immediate user support
- **Assignee:** All Teams | **Effort:** 40h | **Priority:** Critical

## ☐ Post-Launch Optimization

- Monitor user adoption metrics
- Address immediate issues
- Plan next iteration features
- **Assignee:** Product + Engineering | **Effort:** 20h | **Priority:** High

## Resource Allocation

### Team Structure

#### Core Team (6-8 people)

1. **Tech Lead** (1) - Architecture oversight, technical decisions
2. **Frontend Lead** (1) - React development, UI/UX implementation
3. **Backend Lead** (1) - API development, database design
4. **Data Scientist/ML Engineer** (1) - AI algorithms, analytics
5. **DevOps Engineer** (1) - Infrastructure, deployment, monitoring
6. **QA Engineer** (1) - Testing, quality assurance
7. **Frontend Developer** (1-2) - Component development, integration
8. **Backend Developer** (1-2) - API endpoints, business logic

#### Support Team

- **Product Manager** - Requirements, stakeholder communication

- **UX/UI Designer** - Design system, user experience
- **Technical Writer** - Documentation, user guides
- **Security Consultant** - Security audit, compliance

Time Allocation by Phase

Phase	Duration	Effort (Hours)	Cost Estimate
Phase 1: Foundation	2 weeks	320h	\$48,000
Phase 2: Core Features	6 weeks	960h	\$144,000
Phase 3: Enhancement	4 weeks	640h	\$96,000
Phase 4: Testing & QA	4 weeks	640h	\$96,000
Phase 5: Launch	4 weeks	480h	\$72,000
Total	20 weeks	3,040h	\$456,000

*Note: Cost estimates based on average hourly rate of \$150/hour*

Risk Management

High-Risk Items

1. **AI/ML Algorithm Performance** - Mitigation: Early prototyping and validation
2. **Third-Party Integration Complexity** - Mitigation: API testing and fallback plans
3. **Scale Performance Issues** - Mitigation: Load testing and performance monitoring
4. **Data Privacy Compliance** - Mitigation: Legal review and compliance audit

Dependencies

1. **External APIs** - Assessment framework providers
2. **Third-Party Services** - Email, authentication providers
3. **Infrastructure** - Cloud provider availability
4. **Team Availability** - Resource scheduling and backup plans

Success Metrics

Technical KPIs

- **Code Coverage:** >90% for critical paths
- **API Response Time:** <200ms (95th percentile)
- **System Uptime:** >99.9%
- **Page Load Time:** <2 seconds

Business KPIs

- **User Adoption:** 1,000+ active users in first 3 months
- **Assessment Completion Rate:** >85%
- **User Retention:** >70% monthly retention
- **Customer Satisfaction:** >4.5/5 rating

## Quality Gates

### Definition of Done

- ☐ Feature implemented according to specifications
- ☐ Unit tests written and passing (>90% coverage)
- ☐ Integration tests passing
- ☐ Code review completed and approved
- ☐ Security review passed
- ☐ Performance requirements met
- ☐ Documentation updated
- ☐ Acceptance criteria validated

### Release Criteria

- ☐ All critical and high-priority bugs resolved
- ☐ Performance benchmarks met
- ☐ Security audit passed
- ☐ User acceptance testing completed
- ☐ Documentation complete
- ☐ Deployment procedures validated
- ☐ Rollback procedures tested
- ☐ Monitoring and alerting configured

## Communication Plan

### Daily Standups

- **Time:** 9:00 AM daily
- **Duration:** 15 minutes
- **Participants:** Core development team
- **Format:** What I did yesterday, what I'm doing today, blockers

### Sprint Reviews

- **Frequency:** Every 2 weeks
- **Duration:** 2 hours
- **Participants:** Full team + stakeholders

- **Format:** Demo, retrospective, planning

## Stakeholder Updates

- **Frequency:** Weekly
- **Format:** Executive summary email
- **Content:** Progress, risks, upcoming milestones

## Tools & Resources

### Development Tools

- **Version Control:** GitHub Enterprise
- **Project Management:** Jira/Linear
- **Communication:** Slack
- **Documentation:** Confluence/Notion
- **Design:** Figma
- **API Testing:** Postman

### Monitoring Tools

- **Application Monitoring:** DataDog/New Relic
- **Error Tracking:** Sentry
- **Log Aggregation:** ELK Stack
- **Performance Monitoring:** Grafana + Prometheus

## Contingency Plans

### Technical Risks

#### 1. Database Performance Issues

- Plan A: Optimize queries and add indexes
- Plan B: Implement read replicas
- Plan C: Consider database sharding

#### 2. AI/ML Model Accuracy

- Plan A: Fine-tune existing models
- Plan B: Integrate additional data sources
- Plan C: Implement ensemble methods

#### 3. Third-Party Service Failures

- Plan A: Implement retry mechanisms

- Plan B: Use alternative service providers
- Plan C: Build fallback functionality

## **Schedule Risks**

### **1. Resource Unavailability**

- Cross-train team members
- Maintain documentation for knowledge transfer
- Identify backup resources

### **2. Scope Creep**

- Strict change control process
- Regular stakeholder alignment
- Feature prioritization framework

## **Post-Launch Roadmap**

### **Phase 6: Optimization (Weeks 21-24)**

- Performance optimization based on real usage
- User feedback implementation
- A/B testing for key features
- Mobile app development

### **Phase 7: Enterprise Features (Weeks 25-32)**

- Advanced reporting and analytics
- Custom assessment framework builder
- White-label solution
- Advanced integrations (HRIS, ATS)

### **Phase 8: AI Enhancement (Weeks 33-40)**

- Machine learning model improvements
- Predictive analytics enhancement
- Natural language processing features
- Automated insights generation