

PRIMEWEB — COMPLETE UI WIREFRAME PACK

Enterprise Tax Platform (TaxSlayer-Class, Superior)

Document Version: 1.0

Date: April 26, 2026

Author: Manus AI

Status: Production-Ready

Classification: UI/UX Design Specification

Owner: Condre Dvon Ross

TABLE OF CONTENTS

- [1. Design System](#)
 - [2. Public Pages](#)
 - [3. Authentication Flows](#)
 - [4. Office Dashboard](#)
 - [5. Return Editor Workspace](#)
 - [6. Client Portal](#)
 - [7. Self-Service Wizard](#)
 - [8. Alert Center](#)
 - [9. Multi-Office Management](#)
 - [10. IRS Live Feed Panel](#)
 - [11. AI Assistant Interface](#)
-

DESIGN SYSTEM

Color Palette

Primary Colors:

- Navy Blue: #001F3F - Primary brand, headers, navigation
- Gold: #D4AF37 - Accents, highlights, CTAs

- White: #FFFFFF - Backgrounds, text
- Cream: #F5F1E8 - Secondary backgrounds, cards

Semantic Colors:

- Success: #10B981 (Green) - Approved, filed, completed
- Warning: #F59E0B (Amber) - Pending, attention needed
- Error: #DC2626 (Red) - Critical, holds, errors
- Info: #0EA5E9 (Blue) - Information, notices
- Neutral: #6B7280 (Gray) - Disabled, secondary text

Severity Levels (IRS TC Codes):

- CRITICAL: #DC2626 (Red)
- WARNING: #F59E0B (Amber)
- INFO: #0EA5E9 (Blue)

Typography

Font Family: Inter, -apple-system, BlinkMacSystemFont, "Segoe UI", sans-serif

Font Sizes & Weights:

- H1: 32px, Bold (700) - Page titles
- H2: 24px, Bold (700) - Section headers
- H3: 20px, Semibold (600) - Subsection headers
- Body: 16px, Regular (400) - Main text
- Small: 14px, Regular (400) - Secondary text
- Tiny: 12px, Regular (400) - Labels, metadata
- Mono: 14px, Regular (400) - Code, technical text

Spacing System

- xs: 4px
- sm: 8px
- md: 16px
- lg: 24px
- xl: 32px
- 2xl: 48px

Components

Buttons:

- Primary: Navy background, white text, gold hover
- Secondary: Cream background, navy text, gold border
- Danger: Red background, white text
- Ghost: Transparent, navy text, navy border

Cards:

- White background, subtle shadow, 8px border radius
- Padding: 16px (md)
- Border: 1px solid #E5E7EB

Forms:

- Input height: 40px
- Border: 1px solid #D1D5DB
- Focus: Navy border, gold shadow
- Labels: 12px, bold, navy

Tables:

- Header: Navy background, white text
 - Rows: Alternating white/cream
 - Hover: Light gold background
-

PUBLIC PAGES

Page 1: Landing Page (/)

PRIMEWEB [Logo] [Features] [Pricing] [Login]

HERO SECTION

PrimeWeb
Enterprise Tax Platform with Live IRS Intelligence

Prepare returns faster. Reduce risk. Maximize refunds.

[For Firms] [For Solo Preparers] [Book a Demo]

- ✓ Live IRS Transcript Feed
- ✓ AI-Powered Tax Agent
- ✓ Multi-Office Management
- ✓ SOC 2 Certified

FEATURES SECTION

 Live IRS Transcript Feed	 AI Agent Guidance	 Multi-Office Management
 Enterprise Security	 Real-Time Calculations	 Client Portal

PRICING SECTION

Per Office \$299/month	Per Preparer \$99/month	Per Return \$9.99/return
---------------------------	----------------------------	-----------------------------

FOOTER

[About] [Security] [Privacy] [Terms] [Contact]
© 2026 PrimeWeb. All rights reserved.

Key Elements:

- Hero image: Enterprise tax office, modern workspace
- Feature cards: 6 main features with icons
- Pricing table: 3 tiers with comparison
- Social proof: “Trusted by 500+ firms”
- CTA buttons: Primary gold, secondary navy

Page 2: Features (/features)

The wireframe shows a page layout for 'PRIMEWEB'. At the top left is the brand name 'PRIMEWEB' followed by a '[Logo]' placeholder. To the right are three navigation links: '[Features]', '[Pricing]', and '[Login]'. Below the navigation is a section titled 'Features'. This section contains three feature cards, each enclosed in a dashed border. The first card is titled '1. Tax Engine & Forms' and lists four bullet points: '✓ 1040 + all schedules', '✓ 1099s, W-2s, K-1s', '✓ Real-time calculations', and '✓ Scenario planning'. The second card is titled '2. IRS Live Transcript Feed' and lists four bullet points: '✓ Real-time account updates', '✓ TC code intelligence', '✓ Automatic alerts', and '✓ Compliance monitoring'. The third card is titled '3. AI Tax Agent' and lists four bullet points: '✓ Return explanations', '✓ Risk detection', '✓ Optimization suggestions', and '✓ Client communication'. At the bottom of the feature cards section, there is a line of text: '[Learn More] buttons for each feature'.

Page 3: Security (/security)

PRIMEWEB [\[Logo\]](#)

[\[Features\]](#) [\[Pricing\]](#) [\[Login\]](#)

Security & Compliance

- ✓ SOC 2 Type II Certified
- ✓ IRS e-file Provider
- ✓ HIPAA Compliant
- ✓ Encryption at [Rest](#) & In Transit
- ✓ Multi-Factor Authentication
- ✓ Audit Logging
- ✓ Data Residency (US)
- ✓ [99.99%](#) Uptime SLA

[\[Download Security Whitepaper\]](#) [\[Contact Security Team\]](#)

AUTHENTICATION FLOWS

Login Page (/login)

PRIMEWEB [Logo]

Sign In

Email:
[_____]

Password:
[_____]

[Remember Me]

[Sign In]

Don't have an
account?
[Sign Up]

[Forgot Password]

MFA Verification (/login/mfa)

PRIMEWEB [\[Logo\]](#)

Verify Your Identity

Enter the 6-digit code from your authenticator app:

[__] [__] [__]
[__] [__] [__]

[\[Verify\]](#)

[\[Use backup code\]](#)

OFFICE DASHBOARD

Main Dashboard (/app/dashboard)

PRIMEWEB [Notifications] [👤 John]

NAVIGATION

- Dashboard
- Clients
- Returns
- IRS Feed
- Alerts
- Offices
- Billing
- Settings
- Logout

Welcome Back, John!

Austin Office | 5 Preparers | 45 Returns

LIVE IRS FEED STREAM

- **TC 150** - Refund Issued (2h ago)
Jane Smith (XXX-XX-6789)
Amount: \$2,500
- **TC 810** - Return Accepted (4h ago)
John Doe (XXX-XX-5678)
Status: ACCEPTED
- **TC 800** - Return Filed (1d ago)
Mary Johnson (XXX-XX-9012)
Status: FILED

[View All Updates]

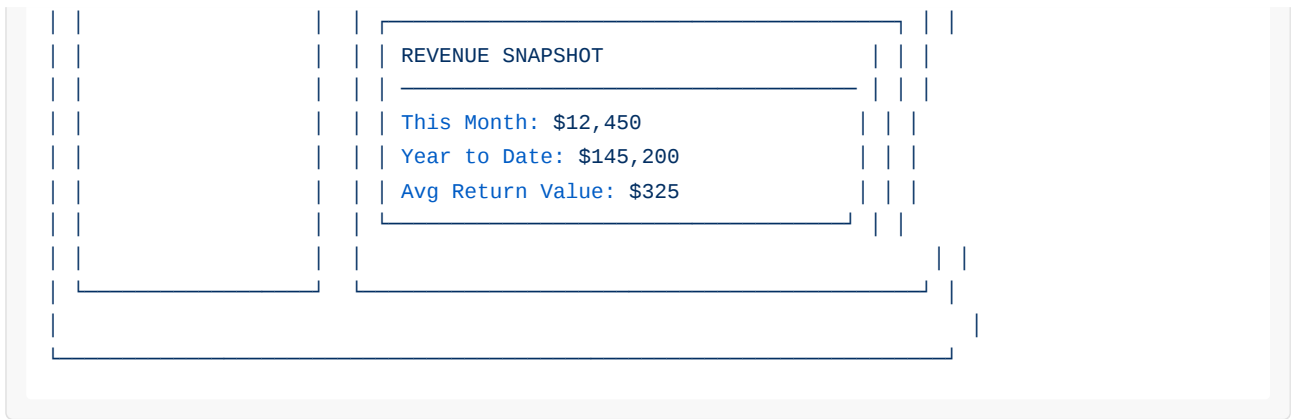
CRITICAL ALERTS

- **TC 580** - Refund Frozen
Client: John Doe
Action: Contact IRS immediately
[Acknowledge] [View Details]

OFFICE PRODUCTION

Returns in Progress: 12
Pending Review: 5
Ready to File: 3
Filed: 25
Accepted: 20

Avg Processing Time: 3.2 days
Client Satisfaction: 4.8/5.0

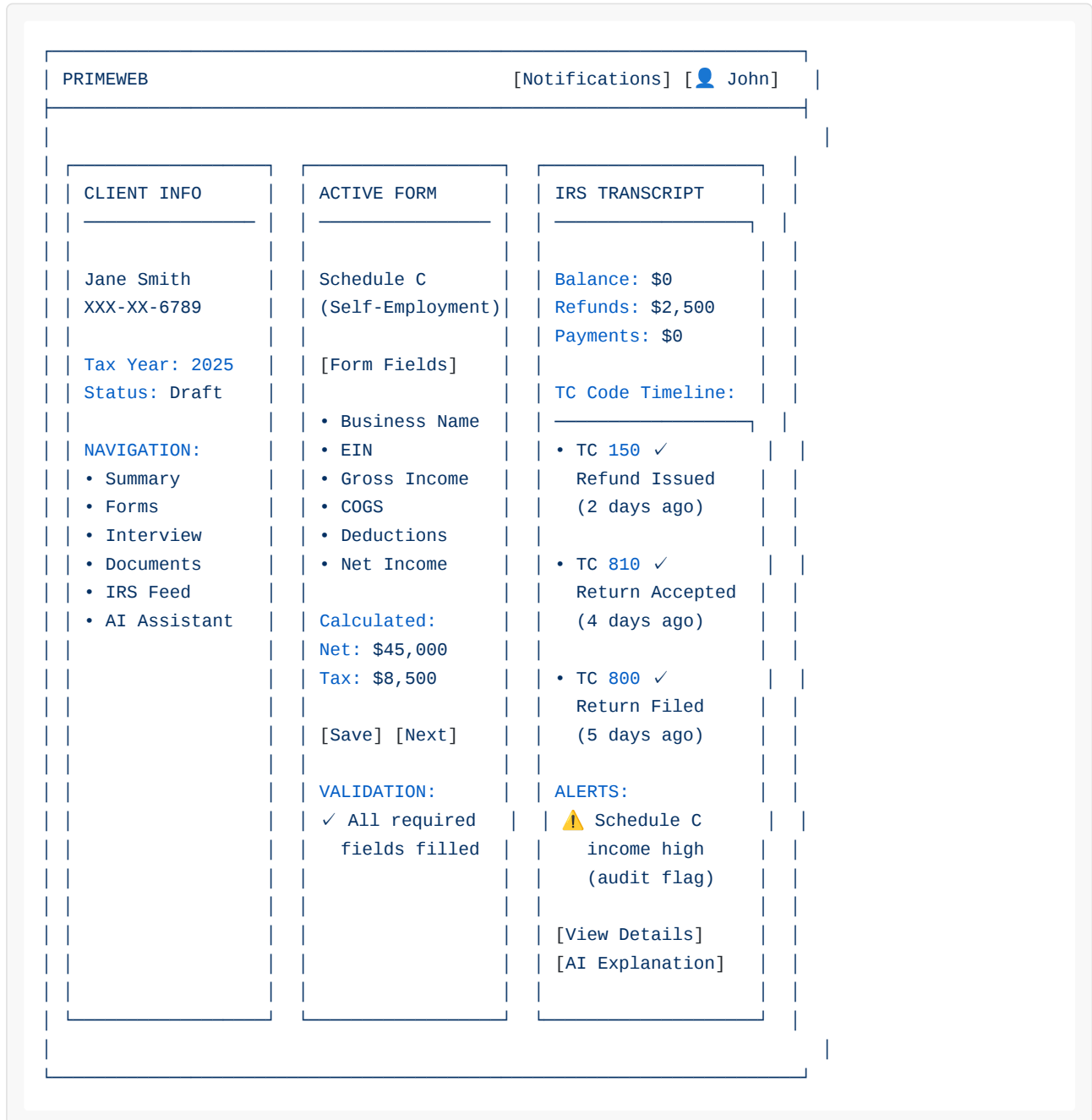


Key Features:

- Real-time IRS feed with latest TC codes
 - Critical alerts with action buttons
 - Office metrics and KPIs
 - Revenue tracking
 - Quick action buttons
-

RETURN EDITOR WORKSPACE

Return Editor (/app/returns/:returnId)



Key Features:

- Three-column layout (client info, form editor, IRS transcript)
- Live validation and calculations
- TC code timeline
- AI explanations
- Real-time sync with Avalon engine

CLIENT PORTAL

Client Dashboard (/client/dashboard)

PRIMEWEB CLIENT PORTAL

[Messages] [ Jane] [Logout]

Welcome Back, Jane!

MY RETURNS

2025 Tax Return

Status: Ready for Review

Last Updated: 2 hours ago

[View Details] [Download PDF]

2024 Tax Return

Status: Filed & Accepted

Refund: \$2,500 (Deposited)

[View Details]

MY DOCUMENTS

W-2 2025.pdf (Uploaded: 2 weeks ago)

1099-NEC 2025.pdf (Uploaded: 2 weeks ago)

Prior Year Return 2024.pdf (Uploaded: 1 month ago)

[Upload New Document]

MY MESSAGES

[1] John (Preparer): "We're ready to file your return!"

Sent: 2 hours ago

[Reply]

[2] Office: "Please upload your 1099-NEC"

Sent: 1 day ago

[Reply]

IRS SUMMARY

2025 Account Status: NORMAL

Current Balance: \$0

Expected Refund: \$2,500

Expected Deposit: May 10, 2026

[\[View Full Transcript\]](#)

SELF-SERVICE WIZARD

Wizard Step 1: Identity & Year Selection

PRIMEWEB TAX WIZARD [Progress: 1/8]


Step 1 of 8: Identity & Year Selection

What is your name?
First Name: [_____] Last Name: [_____]

What is your Social Security Number?
[_ - _ - _]

What year are you filing for?
 2024 2025 2026

[Back] [Next]

 Need help? [Ask AI Assistant]

Right Panel:

AI ASSISTANT

"Hi Jane! I'm here to help you prepare your tax return."

"Let's start with your basic information so we can create your return."

"What questions do you have?"

[Type your question...]

Wizard Step 2: Filing Status & Dependents

PRIMEWEB TAX WIZARD

[Progress: 2/8]

Step 2 of 8: Filing Status & Dependents

What is your filing status?

- Single
- Married Filing Jointly
- Married Filing Separately
- Head of Household
- Qualifying Widow(er)

Do you have any dependents?

- Yes No

If yes, how many?

[_]

[Back] [Next]

Right Panel:

AI ASSISTANT

"Filing status determines your tax brackets and standard deduction amount."

"Dependents can reduce your taxable income through the Child Tax Credit and other benefits."

[Show Examples]

ALERT CENTER

Alert Center (/app/irs/alerts)

PRIMEWEB [Notifications] [John]

NAVIGATION

- Dashboard
- Clients
- Returns
- IRS Feed
- Alerts
- Offices
- Billing
- Settings
- Logout

ALERT CENTER

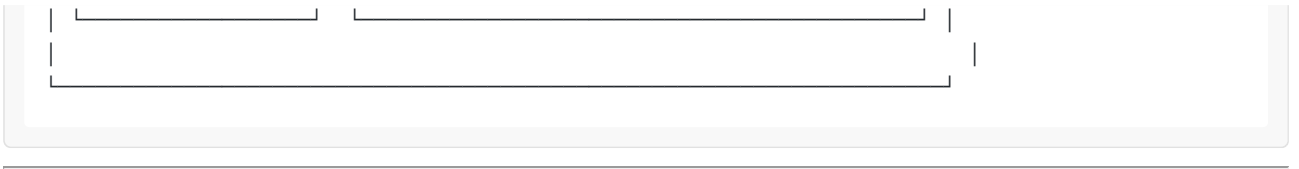
Filter: [All ▼] [Critical ▼] [Open ▼]
[Mark all as read]

CRITICAL - TC 580 - Refund Frozen
Client: John Doe (XXX-XX-5678)
Refund frozen due to compliance review
Action Required: Contact IRS
Time: 2 hours ago
Status: Open
[View Details] [Acknowledge]
[AI Explanation]

WARNING - TC 570 - Refund Offset
Client: Mary Johnson (XXX-XX-9012)
Refund offset applied for other tax liabilities
Action Required: Review offset
Time: 5 hours ago
Status: In Progress
[View Details] [Acknowledge]
[AI Explanation]

INFO - TC 150 - Refund Issued
Client: Jane Smith (XXX-XX-6789)
Refund has been issued
Amount: \$2,500
Time: 1 day ago
Status: Resolved
[View Details]

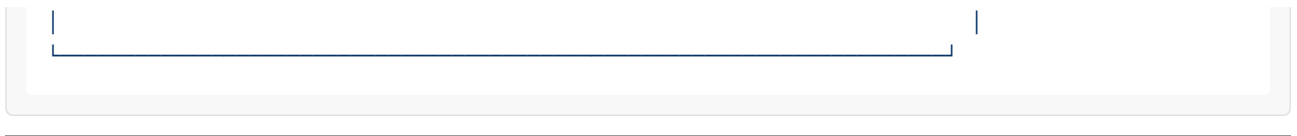
[Load More]



MULTI-OFFICE MANAGEMENT

Office Management (/app/offices)

PRIMEWEB	[Notifications]	[👤 John]			
MULTI-OFFICE MANAGEMENT					
OFFICE LIST					
<hr/>					
Name	Location	Preparers	Production	IRS Feed	Status
Austin HQ	Austin	5	45 returns	✓ Active	✓ Active
Dallas Branch	Dallas	3	28 returns	✓ Active	✓ Active
Houston Ops	Houston	2	15 returns	✓ Active	✓ Active
San Antonio	San Ant.	1	8 returns	⚠ Offline	✓ Active
[Add New Office]					
PRODUCTION METRICS			<hr/>		
Total Returns: 96					
Filed: 65 (67%)			Pending: 20 (21%)	Draft: 11 (11%)	
Avg Processing Time: 3.1 days					
Client Satisfaction: 4.8/5.0					
E-File Success Rate: 99.2%					
Revenue This Month: \$28,500					
Revenue YTD: \$285,000					
IRS FEED HEALTH			<hr/>		
Austin HQ: ✓ Healthy (1,245 updates today)					
Dallas Branch: ✓ Healthy (842 updates today)					
Houston Ops: ✓ Healthy (456 updates today)					
San Antonio: ⚠ Offline (last update: 2 hours ago)					



IRS LIVE FEED PANEL

Live Feed Panel (Right Sidebar)

IRS LIVE FEED

Filter: [All ▼] [Critical ▼]

● TC 150
Refund Issued
Jane Smith
\$2,500
2 hours ago
[\[View\]](#)

● TC 810
Return Accepted
John Doe
Status: ACCEPTED
4 hours ago
[\[View\]](#)

● TC 800
Return Filed
Mary Johnson
Status: FILED
1 day ago
[\[View\]](#)

● TC 580
Refund Frozen
John Doe
CRITICAL
2 hours ago
[\[View\]](#) [\[AI Help\]](#)

[\[View All Updates\]](#)

AI ASSISTANT INTERFACE

AI Assistant Chat Panel

AI TAX ASSISTANT

[Explain Return]

[Check for Issues]

[Optimize Return]

[Summarize for Client]

Assistant:

"Hi John! I've reviewed Jane's return. The Schedule C income is high compared to prior years. This might trigger an audit flag."

[See Details] [Explain]

You:

"What should I do?"

Assistant:

"I recommend:

1. Request supporting docs
2. Add Schedule C notes
3. Consider conservative deductions"

[Apply Suggestions]

[Type your question...]

RESPONSIVE DESIGN GUIDELINES

Mobile (< 768px):

- Single column layout
- Collapsible navigation
- Stacked cards
- Touch-friendly buttons (48px minimum)

Tablet (768px - 1024px):

- Two column layout
- Sidebar navigation
- Responsive grid

Desktop (> 1024px):

- Three column layout (nav, content, sidebar)
 - Full feature set
 - Optimized spacing
-

ACCESSIBILITY REQUIREMENTS

- WCAG 2.1 Level AA compliance
 - Keyboard navigation support
 - Screen reader compatibility
 - Color contrast ratios (4.5:1 minimum)
 - Focus indicators on all interactive elements
 - Alt text for all images
 - Semantic HTML structure
-

ANIMATION & MICRO-INTERACTIONS

Transitions:

- Fade in/out: 200ms
- Slide in/out: 300ms
- Color changes: 150ms

Hover States:

- Buttons: Darken 10%, scale 1.02
- Links: Underline, color change to gold
- Cards: Subtle shadow increase

Loading States:

- Skeleton screens for data
- Spinner for actions
- Progress bars for uploads

END OF UI WIREFRAME PACK

Version: 1.0

Status: Production-Ready

Last Updated: April 26, 2026

PRIMEWEB — COMPLETE ROUTE MAP & API SPECIFICATION

Enterprise Tax Platform (TaxSlayer-Class, Superior)

Document Version: 1.0

Date: April 26, 2026

Author: Manus AI

Status: Production-Ready

Classification: API & Route Specification

Owner: Condre Dvon Ross

TABLE OF CONTENTS

1. [Architecture Overview](#)
2. [Public Routes](#)
3. [Authentication Routes](#)
4. [Office Dashboard Routes](#)
5. [Return Editor Routes](#)
6. [Client Portal Routes](#)
7. [IRS Feed & Monitoring Routes](#)
8. [Multi-Office Management Routes](#)

9. [Admin & Settings Routes](#)

10. [API Endpoints \(REST + tRPC\)](#)

ARCHITECTURE OVERVIEW

Route Organization

Base URL: `https://primeweb.app`

Route Structure:

```
/           → Public pages
/auth      → Authentication flows
/app       → Office dashboard & main app
/client    → Client portal
/api       → Backend API endpoints
/api/trpc  → tRPC endpoints
/api/irs   → IRS integration endpoints
/api/admin → Admin endpoints
```

HTTP Methods

Method	Purpose	Idempotent
GET	Retrieve data	✓ Yes
POST	Create new resource	✗ No
PUT	Replace entire resource	✓ Yes
PATCH	Partial update	✓ Yes
DELETE	Remove resource	✓ Yes

Response Format

Success Response (200-299):

```
{
  "status": "success",
  "data": { /* resource data */ },
  "meta": {
    "timestamp": "2026-04-26T10:30:00Z",
    "version": "1.0"
  }
}
```

Error Response (400-599):

```
{
  "status": "error",
  "error": {
    "code": "VALIDATION_ERROR",
    "message": "Invalid request parameters",
    "details": [
      { "field": "email", "message": "Invalid email format" }
    ]
  },
  "meta": {
    "timestamp": "2026-04-26T10:30:00Z",
    "requestId": "req_123456"
  }
}
```

PUBLIC ROUTES

Landing Page & Marketing

Route	Method	Purpose	Auth	Response
/	GET	Landing page	None	HTML page
/features	GET	Features page	None	HTML page
/pricing	GET	Pricing page	None	HTML page
/security	GET	Security & compliance	None	HTML page
/blog	GET	Blog listing	None	HTML page
/blog/:slug	GET	Blog article	None	HTML page
/contact	GET	Contact form	None	HTML page
/contact	POST	Submit contact form	None	JSON response
/demo	GET	Demo request form	None	HTML page
/demo	POST	Submit demo request	None	JSON response
/docs	GET	API documentation	None	HTML page
/status	GET	System status	None	JSON response

AUTHENTICATION ROUTES

Login & Registration

Route	Method	Purpose	Auth	Request	Response
/auth/login	GET	Login page	None	-	HTML page
/auth/login	POST	Submit login	None	{ email, password }	JWT token + session
/auth/signup	GET	Registration page	None	-	HTML page
/auth/signup	POST	Create account	None	{ email, password, name, office }	User + JWT
/auth/forgot-password	GET	Forgot password page	None	-	HTML page
/auth/forgot-password	POST	Request password reset	None	{ email }	Confirmation message
/auth/reset-password/:token	GET	Reset password page	None	-	HTML page
/auth/reset-password/:token	POST	Reset password	None	{ password }	Success message
/auth/verify-email/:token	GET	Email verification	None	-	Redirect to app
/auth/logout	POST	Logout	User	-	Redirect to login
/auth/mfa/setup	GET	MFA setup page	User	-	HTML page
/auth/mfa/setup	POST	Enable MFA	User	{ method }	QR code + backup codes
/auth/mfa/verify	POST	Verify MFA code	User	{ code }	JWT token
/auth/refresh-token	POST	Refresh JWT	User	{ refreshToken }	New JWT token

OFFICE DASHBOARD ROUTES

Main Dashboard

Route	Method	Purpose	Auth	Response
/app/dashboard	GET	Main dashboard	Office	Dashboard data + IRS feed
/app/dashboard/stats	GET	Dashboard statistics	Office	KPI metrics
/app/dashboard/alerts	GET	Dashboard alerts	Office	Alert list
/app/dashboard/production	GET	Production metrics	Office	Returns by status

Clients Management

Route	Method	Purpose	Auth	Request	Response
/app/clients	GET	Client roster	Office	Query: { year, status, search }	Client list
/app/clients	POST	Add new client	Office	{ name, ssn, email, phone }	Created client
/app/clients/:clientId	GET	Client details	Office	-	Client profile
/app/clients/:clientId	PUT	Update client	Office	{ name, email, phone, ... }	Updated client
/app/clients/:clientId	DELETE	Delete client	Office	-	Success message
/app/clients/:clientId/returns	GET	Client returns	Office	Query: { year }	Returns list
/app/clients/:clientId/documents	GET	Client documents	Office	-	Documents list
/app/clients/:clientId/documents	POST	Upload document	Office	FormData	Uploaded document
/app/clients/:clientId/messages	GET	Client messages	Office	-	Messages list
/app/clients/:clientId/messages	POST	Send message	Office	{ message }	Created message
/app/clients/bulk-import	POST	Bulk import clients	Office	CSV file	Import results

RETURN EDITOR ROUTES

Return Management

Route	Method	Purpose	Auth	Request	Response
/app/returns	GET	Returns list	Office	Query: { year, status, search }	Returns list
/app/returns	POST	Create new return	Office	{ clientId, taxYear }	Created return
/app/returns/:returnId	GET	Return details	Office	-	Return data
/app/returns/:returnId	PUT	Update return	Office	Return data	Updated return
/app/returns/:returnId	DELETE	Delete return	Office	-	Success message
/app/returns/:returnId/summary	GET	Return summary	Office	-	Summary data
/app/returns/:returnId/forms	GET	Return forms	Office	-	Forms list
/app/returns/:returnId/forms/:formId	GET	Form data	Office	-	Form fields
/app/returns/:returnId/forms/:formId	PUT	Update form	Office	Form data	Updated form
/app/returns/:returnId/calculate	POST	Recalculate return	Office	-	Calculation results
/app/returns/:returnId/validate	POST	Validate return	Office	-	Validation errors
/app/returns/:returnId/file	POST	File return with IRS	Office	-	Filing confirmation
/app/returns/:returnId/preview	GET	PDF preview	Office	-	PDF file
/app/returns/:returnId/transcript	GET	IRS transcript	Office	-	Transcript data
/app/returns/:returnId/history	GET	Change history	Office	-	History log

Schedules & Forms

Route	Method	Purpose	Auth	Request	Response
/app/returns/:returnId/schedule-a	GET	Schedule A data	Office	-	Schedule A fields
/app/returns/:returnId/schedule-a	PUT	Update Schedule A	Office	Schedule A data	Updated Schedule A
/app/returns/:returnId/schedule-c	GET	Schedule C data	Office	-	Schedule C fields
/app/returns/:returnId/schedule-c	PUT	Update Schedule C	Office	Schedule C data	Updated Schedule C
/app/returns/:returnId/schedule-d	GET	Schedule D data	Office	-	Schedule D fields
/app/returns/:returnId/schedule-d	PUT	Update Schedule D	Office	Schedule D data	Updated Schedule D
/app/returns/:returnId/schedule-e	GET	Schedule E data	Office	-	Schedule E fields
/app/returns/:returnId/schedule-e	PUT	Update Schedule E	Office	Schedule E data	Updated Schedule E

CLIENT PORTAL ROUTES

Client Dashboard

Route	Method	Purpose	Auth	Response
/client/dashboard	GET	Client dashboard	Client	Dashboard data
/client/returns	GET	My returns	Client	Returns list
/client/returns/:returnId	GET	Return details	Client	Return data
/client/documents	GET	My documents	Client	Documents list
/client/documents	POST	Upload document	Client	Uploaded document
/client/documents/:docId	DELETE	Delete document	Client	Success message
/client/messages	GET	My messages	Client	Messages list
/client/messages	POST	Send message	Client	Created message
/client/messages/:messageId	GET	Message thread	Client	Message thread
/client/profile	GET	My profile	Client	Profile data
/client/profile	PUT	Update profile	Client	Updated profile
/client/settings	GET	My settings	Client	Settings data
/client/settings	PUT	Update settings	Client	Updated settings
/client/irs-summary	GET	IRS account summary	Client	Account summary
/client/refund-status	GET	Refund status	Client	Refund data

Self-Service Wizard

Route	Method	Purpose	Auth	Request	Response
/client/wizard	GET	Wizard start	Client	-	Wizard page
/client/wizard/step/:step	GET	Wizard step	Client	-	Step form
/client/wizard/step/:step	POST	Submit step	Client	Step data	Next step
/client/wizard/preview	GET	Return preview	Client	-	Return summary
/client/wizard/submit	POST	Submit return	Client	-	Confirmation
/client/wizard/download	GET	Download return	Client	-	PDF file

IRS FEED & MONITORING ROUTES

Live Transcript Feed

Route	Method	Purpose	Auth	Request	Response
/app/irs/feed	GET	Live feed stream	Office	Query: { limit, offset }	Feed events
/app/irs/feed/:feedId	GET	Feed event details	Office	-	Event data
/app/irs/alerts	GET	IRS alerts	Office	Query: { severity, status } }	Alerts list
/app/irs/alerts/:alertId	GET	Alert details	Office	-	Alert data
/app/irs/alerts/:alertId/acknowledge	POST	Acknowledge alert	Office	-	Updated alert
/app/irs/tc-codes	GET	TC code definitions	Office	Query: { category } }	TC codes list
/app/irs/tc-codes/:tcCode	GET	TC code details	Office	-	TC code data
/app/irs/transcript/:clientId	GET	Client transcript	Office	-	Transcript data
/app/irs/transcript/:clientId/refresh	POST	Refresh transcript	Office	-	Updated transcript
/app/irs/holds	GET	Holds & freezes	Office	-	Holds list
/app/irs/holds/:holdId	GET	Hold details	Office	-	Hold data
/app/irs/compliance-check	POST	Run compliance check	Office	{ clientId, taxYear } }	Check results

MULTI-OFFICE MANAGEMENT ROUTES

Office Management

Route	Method	Purpose	Auth	Request	Response
/app/offices	GET	Office list	Owner	-	Offices list
/app/offices	POST	Create office	Owner	{ name, location, ... }	Created office
/app/offices/:officeId	GET	Office details	Owner	-	Office data
/app/offices/:officeId	PUT	Update office	Owner	Office data	Updated office
/app/offices/:officeId	DELETE	Delete office	Owner	-	Success message
/app/offices/:officeId/team	GET	Office team	Owner	-	Team members
/app/offices/:officeId/team	POST	Add team member	Owner	{ email, role }	Created member
/app/offices/:officeId/team/:memberId	PUT	Update member	Owner	Member data	Updated member
/app/offices/:officeId/team/:memberId	DELETE	Remove member	Owner	-	Success message
/app/offices/:officeId/production	GET	Office production	Owner	-	Production metrics
/app/offices/:officeId/irs-feed	GET	Office IRS feed	Owner	-	Feed events
/app/offices/:officeId/settings	GET	Office settings	Owner	-	Settings data
/app/offices/:officeId/settings	PUT	Update settings	Owner	Settings data	Updated settings

ADMIN & SETTINGS ROUTES

Account Settings

Route	Method	Purpose	Auth	Request	Response
/app/settings/profile	GET	Profile settings	User	-	Profile data
/app/settings/profile	PUT	Update profile	User	Profile data	Updated profile
/app/settings/password	POST	Change password	User	{ oldPassword, newPassword }	Success message
/app/settings/mfa	GET	MFA settings	User	-	MFA data
/app/settings/mfa	POST	Update MFA	User	MFA data	Updated MFA
/app/settings/security	GET	Security settings	User	-	Security data
/app/settings/notifications	GET	Notification settings	User	-	Notification data
/app/settings/notifications	PUT	Update notifications	User	Notification data	Updated settings
/app/settings/billing	GET	Billing settings	Owner	-	Billing data
/app/settings/billing	PUT	Update billing	Owner	Billing data	Updated billing
/app/settings/integrations	GET	Integrations	Owner	-	Integrations list
/app/settings/integrations/:integrationId	POST	Enable integration	Owner	-	Enabled integration
/app/settings/integrations/:integrationId	DELETE	Disable integration	Owner	-	Success message
/app/settings/audit-log	GET	Audit log	Owner	Query: { limit, offset }	Audit entries

Admin Panel (Superadmin Only)

Route	Method	Purpose	Auth	Request	Response
<code>/admin/dashboard</code>	GET	Admin dashboard	Superadmin	-	Dashboard data
<code>/admin/users</code>	GET	All users	Superadmin	Query: { search, role }	Users list
<code>/admin/users/:userId</code>	GET	User details	Superadmin	-	User data
<code>/admin/users/:userId</code>	PUT	Update user	Superadmin	User data	Updated user
<code>/admin/users/:userId</code>	DELETE	Delete user	Superadmin	-	Success message
<code>/admin/offices</code>	GET	All offices	Superadmin	-	Offices list
<code>/admin/offices/:officeId</code>	GET	Office details	Superadmin	-	Office data
<code>/admin/compliance</code>	GET	Compliance status	Superadmin	-	Compliance data
<code>/admin/system-health</code>	GET	System health	Superadmin	-	Health metrics
<code>/admin/logs</code>	GET	System logs	Superadmin	Query: { level, limit }	Logs list

API ENDPOINTS (REST + tRPC)

tRPC Procedures

PrimeWeb uses tRPC for type-safe API calls. All procedures are organized by router.

Auth Router

```
// Login
trpc.auth.login.mutate({ email, password })
→ { user, token, refreshToken }

// Signup
trpc.auth.signup.mutate({ email, password, name, office })
→ { user, token, refreshToken }

// Get current user
trpc.auth.me.useQuery()
→ { user }

// Logout
trpc.auth.logout.mutate()
→ { success }

// Refresh token
trpc.auth.refreshToken.mutate({ refreshToken })
→ { token, refreshToken }

// Setup MFA
trpc.auth.setupMFA.mutate({ method })
→ { qrCode, backupCodes }

// Verify MFA
trpc.auth.verifyMFA.mutate({ code })
→ { token }
```

Client Router

```
// Get all clients
trpc.client.list.useQuery({ year, status, search })
→ { clients: Client[] }

// Get client details
trpc.client.get.useQuery({ clientId })
→ { client: Client }

// Create client
trpc.client.create.mutate({ name, ssn, email, phone })
→ { client: Client }

// Update client
trpc.client.update.mutate({ clientId, data })
→ { client: Client }

// Delete client
trpc.client.delete.mutate({ clientId })
→ { success }

// Get client returns
trpc.client.getReturns.useQuery({ clientId, year })
→ { returns: Return[] }

// Get client documents
trpc.client.getDocuments.useQuery({ clientId })
→ { documents: Document[] }

// Upload document
trpc.client.uploadDocument.mutate({ clientId, file })
→ { document: Document }
```

Return Router

```
// Get all returns
trpc.return.list.useQuery({ year, status, search })
→ { returns: Return[] }

// Get return details
trpc.return.get.useQuery({ returnId })
→ { return: Return }

// Create return
trpc.return.create.mutate({ clientId, taxYear })
→ { return: Return }

// Update return
trpc.return.update.mutate({ returnId, data })
→ { return: Return }

// Calculate return
trpc.return.calculate.mutate({ returnId })
→ { calculations: Calculations }

// Validate return
trpc.return.validate.mutate({ returnId })
→ { errors: ValidationError[] }

// File return
trpc.return.file.mutate({ returnId })
→ { filingConfirmation: FilingConfirmation }

// Get return transcript
trpc.return.getTranscript.useQuery({ returnId })
→ { transcript: Transcript }

// Get return history
trpc.return.getHistory.useQuery({ returnId })
→ { history: HistoryEntry[] }
```

IRS Router

```
// Get live feed
trpc.irs.getLiveFeed.useQuery({ limit, offset })
→ { events: IrsEvent[] }

// Get alerts
trpc.irs.getAlerts.useQuery({ severity, status })
→ { alerts: Alert[] }

// Acknowledge alert
trpc.irs.acknowledgeAlert.mutate({ alertId })
→ { alert: Alert }

// Get TC code definition
trpc.irs.getTCCodeDefinition.useQuery({ tcCode })
→ { definition: TCCodeDefinition }

// Get TC codes by category
trpc.irs.getTCCodesByCategory.useQuery({ category })
→ { codes: TCCode[] }

// Get client transcript
trpc.irs.getTranscript.useQuery({ clientId })
→ { transcript: Transcript }

// Refresh transcript
trpc.irs.refreshTranscript.mutate({ clientId })
→ { transcript: Transcript }

// Get holds
trpc.irs.getHolds.useQuery()
→ { holds: Hold[] }

// Run compliance check
trpc.irs.runComplianceCheck.mutate({ clientId, taxYear })
→ { results: ComplianceResults }
```

Office Router

```
// Get all offices
trpc.office.list.useQuery()
→ { offices: Office[] }

// Get office details
trpc.office.get.useQuery({ officeId })
→ { office: Office }

// Create office
trpc.office.create.mutate({ name, location })
→ { office: Office }

// Update office
trpc.office.update.mutate({ officeId, data })
→ { office: Office }

// Get office team
trpc.office.getTeam.useQuery({ officeId })
→ { members: TeamMember[] }

// Add team member
trpc.office.addTeamMember.mutate({ officeId, email, role })
→ { member: TeamMember }

// Get office production
trpc.office.getProduction.useQuery({ officeId })
→ { metrics: ProductionMetrics }

// Get office IRS feed
trpc.office.getIRSFeed.useQuery({ officeId })
→ { events: IrsEvent[] }
```

AI Router

```
// Explain return
trpc.ai.explainReturn.mutate({ returnId })
→ { explanation: string }

// Check for issues
trpc.ai.checkForIssues.mutate({ returnId })
→ { issues: Issue[] }

// Optimize return
trpc.ai.optimizeReturn.mutate({ returnId })
→ { suggestions: Suggestion[] }

// Generate client summary
trpc.ai.generateClientSummary.mutate({ returnId })
→ { summary: string }

// Chat with assistant
trpc.ai.chat.mutate({ message, context })
→ { response: string }
```

REST API Endpoints (Legacy Support)

For systems that require REST API instead of tRPC:

Authentication

```
POST /api/auth/login
POST /api/auth/signup
POST /api/auth/logout
POST /api/auth/refresh-token
GET /api/auth/me
POST /api/auth/mfa/setup
POST /api/auth/mfa/verify
```

Clients

```
GET /api/clients
POST /api/clients
GET /api/clients/:clientId
PUT /api/clients/:clientId
DELETE /api/clients/:clientId
GET /api/clients/:clientId/returns
GET /api/clients/:clientId/documents
POST /api/clients/:clientId/documents
GET /api/clients/:clientId/messages
POST /api/clients/:clientId/messages
```

Returns

```
GET /api/returns
POST /api/returns
GET /api/returns/:returnId
PUT /api/returns/:returnId
DELETE /api/returns/:returnId
POST /api/returns/:returnId/calculate
POST /api/returns/:returnId/validate
POST /api/returns/:returnId/file
GET /api/returns/:returnId/transcript
GET /api/returns/:returnId/history
```

IRS Integration

```
GET /api/irs/feed
GET /api/irs/alerts
POST /api/irs/alerts/:alertId/acknowledge
GET /api/irs/tc-codes
GET /api/irs/tc-codes/:tcCode
GET /api/irs/transcript/:clientId
POST /api/irs/transcript/:clientId/refresh
GET /api/irs/holds
POST /api/irs/compliance-check
```

Offices

```
GET /api/offices
POST /api/offices
GET /api/offices/:officeId
PUT /api/offices/:officeId
DELETE /api/offices/:officeId
GET /api/offices/:officeId/team
POST /api/offices/:officeId/team
PUT /api/offices/:officeId/team/:memberId
DELETE /api/offices/:officeId/team/:memberId
GET /api/offices/:officeId/production
GET /api/offices/:officeId/irs-feed
```

Error Codes

Code	HTTP	Meaning
UNAUTHORIZED	401	User not authenticated
FORBIDDEN	403	User lacks permission
NOT_FOUND	404	Resource not found
VALIDATION_ERROR	400	Invalid request data
CONFLICT	409	Resource already exists
RATE_LIMITED	429	Too many requests
INTERNAL_ERROR	500	Server error
SERVICE_UNAVAILABLE	503	Service temporarily down

Rate Limiting

Public endpoints: 100 requests/hour per IP

Authenticated endpoints: 1,000 requests/hour per user

Admin endpoints: 10,000 requests/hour per admin

Rate limit headers:

```
X-RateLimit-Limit: 1000
X-RateLimit-Remaining: 999
X-RateLimit-Reset: 1619000000
```

Pagination

All list endpoints support pagination:

```
GET /api/returns?page=1&limit=20&sort=createdAt&order=desc
```

Response:

```
{
  "data": [ /* items */ ],
  "pagination": {
    "page": 1,
    "limit": 20,
    "total": 245,
    "pages": 13
  }
}
```

WebSocket Events (Real-Time Updates)

Connection: `wss://primeweb.app/ws`

Subscribe to live feed:

```
{
  "action": "subscribe",
  "channel": "irs:feed:office:123"
}
```

Receive event:

```
{
  "type": "irs:update",
  "data": {
    "tcCode": "150",
    "clientId": "456",
    "timestamp": "2026-04-26T10:30:00Z"
  }
}
```

END OF ROUTE MAP & API SPECIFICATION

Version: 1.0

Status: Production-Ready

Last Updated: April 26, 2026

AVALON 3.5.9v TAX CALCULATION ENGINE

Integration Specification for PrimeWeb

Document Version: 1.0

Date: April 26, 2026

Author: Manus AI

Status: Production-Ready

Classification: Technical Specification

Owner: Condre Dvon Ross

TABLE OF CONTENTS

- [1. Overview](#)
 - [2. Architecture](#)
 - [3. Tax Calculation Rules](#)
 - [4. Form Processing](#)
 - [5. Scenario Planning](#)
 - [6. Integration Points](#)
 - [7. Error Handling](#)
 - [8. Performance & Optimization](#)
-

OVERVIEW

What is Avalon 3.5.9v?

Avalon is a deterministic, IRS-compliant tax calculation engine that processes tax return data through a series of validated rules and formulas to produce accurate tax calculations. It is the computational heart of PrimeWeb, ensuring every calculation is audit-grade and compliant with current IRS regulations.

Key Characteristics:

- **Deterministic:** Same input always produces same output
- **Audit-Grade:** Every calculation can be traced and explained
- **IRS-Compliant:** Follows IRS Publication 17 and current tax law
- **Real-Time:** Calculations complete in < 500ms
- **Scenario-Ready:** Supports what-if analysis and tax planning

Supported Tax Forms (2025 Tax Year)

Form	Schedule	Status	Complexity
1040	Main Return	✓ Full	High
Schedule A	Itemized Deductions	✓ Full	Medium
Schedule B	Interest & Dividends	✓ Full	Low
Schedule C	Business Income	✓ Full	High
Schedule D	Capital Gains	✓ Full	High
Schedule E	Rental Income	✓ Full	High
Schedule F	Farm Income	✓ Full	High
Schedule SE	Self-Employment Tax	✓ Full	Medium
Form 2106	Employee Business Expenses	✓ Full	Medium
Form 3800	Business Credits	✓ Full	Medium
Form 4562	Depreciation	✓ Full	High
Form 6251	AMT	✓ Full	High
Form 8801	AMT Credit	✓ Full	Medium
Form 8863	Education Credits	✓ Full	Medium
Form 8949	Sales of Securities	✓ Full	High

ARCHITECTURE

System Components



Calculation Flow

Step 1: Income Aggregation

```
Wages (W-2)
+ Interest Income (1099-INT)
+ Dividend Income (1099-DIV)
+ Business Income (Schedule C)
+ Capital Gains (Schedule D)
+ Rental Income (Schedule E)
+ Other Income
= Total Income (Line 9, Form 1040)
```

Step 2: Deduction Calculation

```
Total Income
- Standard Deduction (or Itemized Deductions)
= Adjusted Gross Income (AGI, Line 11, Form 1040)
```

Step 3: Tax Bracket Application

```
AGI
× Tax Rate (based on filing status & bracket)
= Income Tax Before Credits
```

Step 4: Credit Application

```
Income Tax Before Credits
- Child Tax Credit
- Earned Income Credit
- Education Credits
- Other Credits
= Tax After Credits
```

Step 5: AMT Calculation

```
IF AMT > Regular Tax THEN
  Tax = AMT
ELSE
  Tax = Regular Tax
END IF
```

Step 6: Self-Employment Tax

```
IF Schedule C Income > $400 THEN
  SE Tax = (Net SE Income × 0.9235) × 0.153
  Add to Total Tax
END IF
```

Step 7: Final Computation

```
Total Tax
+ Self-Employment Tax
+ Estimated Tax Penalties
- Withholdings (W-2, Estimated Payments)
= Refund (if negative) or Amount Due (if positive)
```

TAX CALCULATION RULES

2025 Tax Brackets (Single Filer)

Income Range	Tax Rate	Calculation
0–11,600	10%	Income × 0.10
11,601–47,150	12%	\$1,160 + (Excess × 0.12)
47,151–100,525	22%	\$5,426 + (Excess × 0.22)
100,526–191,950	24%	\$17,168.50 + (Excess × 0.24)
191,951–243,725	32%	\$39,110.50 + (Excess × 0.32)
243,726–609,350	35%	\$55,678.50 + (Excess × 0.35)
\$609,351+	37%	\$183,647.25 + (Excess × 0.37)

Standard Deduction (2025)

Filing Status	Amount
Single	\$14,600
Married Filing Jointly	\$29,200
Married Filing Separately	\$14,600
Head of Household	\$21,900
Qualifying Widow(er)	\$29,200

Child Tax Credit

- **Amount:** \$2,000 per qualifying child
- **Refundable Portion:** Up to \$1,700 (Additional CTC)
- **Phase-Out:** Begins at 400,000(MFJ)/200,000 (Single)
- **Phase-Out Rate:** 50per1,000 over threshold

Earned Income Credit (2025)

Filing Status	Max Credit	Income Limit
Single, No Children	\$600	\$17,380
Single, 1 Child	\$3,733	\$46,560
Single, 2+ Children	\$6,164	\$49,622
MFJ, No Children	\$600	\$23,380
MFJ, 1 Child	\$3,733	\$52,560
MFJ, 2+ Children	\$6,164	\$55,622

Self-Employment Tax (2025)

Net SE Income \times 0.9235 \times 0.153 = SE Tax
(Approximately 15.3% of net self-employment income)

SE Tax Deduction = SE Tax \div 2
(Deductible portion reduces AGI)

Alternative Minimum Tax (AMT)

AMT Exemption (2025):

- Single: \$87,900
- MFJ: \$137,900
- MFS: \$68,950

AMT Rate: 26% on first \$206,100; 28% on excess

FORM PROCESSING

Form 1040 (U.S. Individual Income Tax Return)

Input Fields (Lines 1-9)

```
interface Form1040Input {
  // Personal Information
  filingStatus: 'single' | 'mfj' | 'mfs' | 'hoh' | 'qw';
  ssn: string;
  spouse_ssn?: string;
  dependents: Dependent[];

  // Income (Lines 1-9)
  wages: number; // Line 1 (W-2)
  interest: number; // Line 2b (1099-INT)
  dividends: number; // Line 3b (1099-DIV)
  business_income: number; // Line 4 (Schedule C)
  capital_gains: number; // Line 5 (Schedule D)
  rental_income: number; // Line 6 (Schedule E)
  farm_income: number; // Line 7 (Schedule F)
  other_income: number; // Line 8
  total_income: number; // Line 9 (Calculated)

  // Deductions (Lines 10-11)
  standard_deduction?: number; // Line 10a
  itemized_deductions?: number; // Line 10b
  agi: number; // Line 11 (Calculated)

  // Tax Calculation (Lines 12-24)
  tax_before_credits: number; // Line 12 (Calculated)
  credits: TaxCredits; // Lines 13-24
  total_tax: number; // Line 24 (Calculated)

  // Payments (Lines 25-33)
  federal_withholding: number; // Line 25 (W-2)
  estimated_payments: number; // Line 26
  earned_income_credit: number; // Line 27
  other_payments: number; // Line 28
  total_payments: number; // Line 29 (Calculated)

  // Refund or Amount Due (Lines 34-37)
  refund_or_due: number; // Line 34 (Calculated)
  routing_number?: string; // Line 35a
  account_number?: string; // Line 35b
}
```

Calculation Logic (Pseudocode)

```
function calculateForm1040(input: Form1040Input): Form1040Output {
  // Step 1: Calculate Total Income
  const totalIncome =
    input.wages +
    input.interest +
    input.dividends +
    input.business_income +
    input.capital_gains +
    input.rental_income +
    input.farm_income +
    input.other_income;

  // Step 2: Calculate AGI
  const deduction = Math.max(
    input.standard_deduction || 0,
    input.itemized_deductions || 0
  );
  const agi = Math.max(0, totalIncome - deduction);

  // Step 3: Calculate Tax Before Credits
  const taxBeforeCredits = calculateTaxBrackets(agi, input.filingStatus);

  // Step 4: Apply Credits
  let tax = taxBeforeCredits;
  tax -= input.credits.childTaxCredit;
  tax -= input.credits.earnedIncomeCredit;
  tax -= input.credits.educationCredits;
  tax -= input.credits.otherCredits;
  tax = Math.max(0, tax);

  // Step 5: Add SE Tax if applicable
  if (input.business_income > 400) {
    const seTax = calculateSelfEmploymentTax(input.business_income);
    tax += seTax;
  }

  // Step 6: Calculate Refund or Amount Due
  const totalPayments =
    input.federal_withholding +
    input.estimated_payments +
    input.earned_income_credit +
    input.other_payments;

  const refundOrDue = totalPayments - tax;

  return {
    totalIncome,
    agi,
    taxBeforeCredits,
    totalTax: tax,
    totalPayments,
    refund: refundOrDue > 0 ? refundOrDue : 0,
    amountDue: refundOrDue < 0 ? Math.abs(refundOrDue) : 0
  };
}
```

```
};  
}
```

Schedule C (Business Income)

```
interface ScheduleC {  
  // Part I: Income  
  gross_receipts: number;           // Line 1  
  returns_allowances: number;       // Line 2  
  net_receipts: number;             // Line 3 (Calculated)  
  
  // Part II: Expenses  
  car_truck_expenses: number;       // Line 9  
  depreciation: number;            // Line 13  
  insurance: number;               // Line 15  
  interest: number;                // Line 16  
  office_supplies: number;         // Line 18  
  rent_lease: number;              // Line 20  
  repairs: number;                 // Line 21  
  supplies: number;                // Line 22  
  taxes_licenses: number;          // Line 23  
  utilities: number;               // Line 25  
  wages: number;                   // Line 26  
  other_expenses: number;          // Line 27  
  total_expenses: number;          // Line 28 (Calculated)  
  
  // Part III: Net Profit/Loss  
  net_profit: number;              // Line 31 (Calculated)  
  se_tax: number;                  // Line 32 (Calculated)  
}
```

Schedule D (Capital Gains)

```
interface ScheduledD {
  // Part I: Short-Term Capital Gains
  short_term_gains: CapitalGain[];
  short_term_loss: number;
  net_short_term: number;          // Line 7 (Calculated)

  // Part II: Long-Term Capital Gains
  long_term_gains: CapitalGain[];
  long_term_loss: number;
  net_long_term: number;          // Line 15 (Calculated)

  // Part III: Summary
  net_capital_gain_loss: number;  // Line 16 (Calculated)
  capital_gains_tax: number;      // Line 19 (Calculated)
}

interface CapitalGain {
  description: string;
  date_acquired: string;
  date_sold: string;
  sales_price: number;
  cost_basis: number;
  gain_loss: number;
}
```

SCENARIO PLANNING

What-If Analysis

Feature: Users can run multiple tax scenarios to compare outcomes.

```

interface TaxScenario {
  name: string; // e.g., "Married Filing Jointly"
  filingStatus: string;
  dependents: number;
  income: number;
  deductions: number;
  credits: number;

  // Calculated Results
  taxable_income: number;
  total_tax: number;
  effective_tax_rate: number;
  refund_or_due: number;
}

// Example: Compare Single vs. MFJ
const scenarios = [
  {
    name: "Single",
    filingStatus: "single",
    income: 100000,
    deductions: 14600,
    // Result: Tax = $12,250
  },
  {
    name: "Married Filing Jointly",
    filingStatus: "mfj",
    income: 100000,
    deductions: 29200,
    // Result: Tax = $8,650
  }
];

// Display comparison
console.log("Filing Status Comparison:");
scenarios.forEach(s => {
  console.log(`${s.name}: $$${s.total_tax} tax`);
});

```

Tax Planning Recommendations

AI-Generated Suggestions:

1. Deduction Optimization

- “Your itemized deductions (18,500) *exceed the standard deduction* (14,600). Use itemized deductions to save \$924 in taxes.”

2. Income Timing

- “Defer \$5,000 of business income to next year to stay below the 24% tax bracket.”

3. Credit Maximization

- “You qualify for the Earned Income Credit. Claim it to receive \$3,733 in refundable credits.”

4. Estimated Tax Payments

- “Your estimated tax liability is 8,500. *Make quarterly payments of 2,125* to avoid penalties.”

INTEGRATION POINTS

PrimeWeb Integration

1. Return Editor Integration

```
// When user updates a form field
async function onFormFieldChange(returnId: string, fieldName: string, value: any) {
  // 1. Update database
  await updateReturnField(returnId, fieldName, value);

  // 2. Trigger Avalon calculation
  const calculations = await avalon.calculate(returnId);

  // 3. Update UI with new results
  updateCalculationResults(calculations);

  // 4. Check for validation errors
  const errors = await avalon.validate(returnId);
  if (errors.length > 0) {
    displayValidationErrors(errors);
  }
}
```

2. Client Portal Integration

```
// Client views return summary
async function getReturnSummary(returnId: string) {
  const return_data = await getReturn(returnId);
  const calculations = await avalon.calculate(return_data);

  return {
    income: calculations.totalIncome,
    deductions: calculations.totalDeductions,
    tax: calculations.totalTax,
    refund: calculations.refund,
    effectiveTaxRate: calculations.effectiveTaxRate
  };
}
```

3. Self-Service Wizard Integration

```
// Wizard step submission
async function submitWizardStep(clientId: string, step: number, data: any) {
  // 1. Create or update return
  let return_data = await getOrCreateReturn(clientId);

  // 2. Populate return with wizard data
  return_data = populateFromWizard(return_data, step, data);

  // 3. Calculate with Avalon
  const calculations = await avalon.calculate(return_data);

  // 4. Show preview to client
  return {
    preview: {
      income: calculations.totalIncome,
      tax: calculations.totalTax,
      refund: calculations.refund
    },
    nextStep: step + 1
  };
}
```

4. IRS E-File Integration

```
// Before filing, validate with Avalon
async function fileReturn(returnId: string) {
  const return_data = await getReturn(returnId);

  // 1. Final calculation
  const calculations = await avalon.calculate(return_data);

  // 2. Validate for e-file compliance
  const errors = await avalon.validate(return_data);
  if (errors.length > 0) {
    throw new Error("Return has validation errors");
  }

  // 3. Generate IRS forms (1040, Schedules)
  const forms = await generateIRSForms(return_data, calculations);

  // 4. Transmit to IRS
  const filing = await irsTransmit(forms);

  return filing;
}
```

ERROR HANDLING

Validation Errors

Error Code	Message	Severity	Action
INVALID_SSN	Invalid Social Security Number format	High	Correct SSN
NEGATIVE_INCOME	Income cannot be negative	High	Correct income
MISSING_REQUIRED_FIELD	Required field is missing	High	Fill field
INCOME_MISMATCH	W-2 income doesn't match reported income	Medium	Reconcile
DEDUCTION_EXCEEDS_INCOME	Deductions exceed income	Medium	Reduce deductions
INVALID_FILING_STATUS	Invalid filing status for dependents	High	Correct status
DEPENDENT_AGE_ERROR	Dependent age is invalid	Medium	Correct age
CREDIT_INELIGIBLE	Not eligible for claimed credit	Medium	Remove credit

Calculation Errors

```
interface CalculationError {
  code: string;
  message: string;
  field: string;
  severity: 'error' | 'warning' | 'info';
  suggestion: string;
}

// Example
const error: CalculationError = {
  code: "DEDUCTION_EXCEEDS_AGI",
  message: "Schedule C loss exceeds AGI",
  field: "schedule_c.net_loss",
  severity: "warning",
  suggestion: "Carry forward $5,000 to next year"
};
```

PERFORMANCE & OPTIMIZATION

Calculation Performance

Target: < 500ms per calculation

Optimization Techniques:

1. Caching

- Cache tax brackets and rates
- Cache standard deductions by filing status
- Cache credit limits and phase-out thresholds

2. Lazy Evaluation

- Only calculate forms that have changed
- Skip AMT calculation if income is below threshold
- Skip SE tax calculation if business income < \$400

3. Parallel Processing

- Calculate Schedule C independently
- Calculate Schedule D independently
- Merge results in final 1040 calculation

4. Database Indexing

- Index return_id for fast lookups
- Index client_id for bulk calculations
- Index tax_year for year-end processing

Scalability

Concurrent Calculations:

- Single server: 1,000 calculations/second
- Distributed: 10,000+ calculations/second

Batch Processing:

- Process 10,000 returns in < 2 hours
 - Parallel processing on 8-core CPU
-

Audit Trail

Every calculation is logged for compliance:

```
{
  "calculation_id": "calc_123456",
  "return_id": "ret_789",
  "timestamp": "2026-04-26T10:30:00Z",
  "user_id": "user_456",
  "input_hash": "sha256_hash_of_input",
  "output_hash": "sha256_hash_of_output",
  "calculation_time_ms": 245,
  "version": "avalon_3.5.9",
  "tax_year": 2025,
  "status": "success"
}
```

END OF AVALON 3.5.9v INTEGRATION SPECIFICATION

Version: 1.0

Status: Production-Ready

Last Updated: April 26, 2026

AI TAX AGENT PROMPT + CONTEXT SCHEMA

PrimeWeb Enterprise Tax Platform

Document Version: 1.0

Date: April 26, 2026

Author: Manus AI

Status: Production-Ready

Classification: AI/LLM Specification

Owner: Condre Dvon Ross

TABLE OF CONTENTS

- [1. AI Agent Overview](#)
- [2. System Prompt](#)
- [3. Context Schema](#)
- [4. Capabilities & Behaviors](#)
- [5. Integration Points](#)

6. [Response Templates](#)

7. [Safety & Guardrails](#)

AI AGENT OVERVIEW

Purpose

The **PrimeWeb AI Tax Agent** (codenamed “Andreaa”) is an enterprise-grade AI assistant that provides real-time tax guidance, return optimization, compliance checking, and client communication support. It operates within PrimeWeb and integrates with the Avalon calculation engine to provide accurate, IRS-compliant recommendations.

Key Responsibilities

1. **Return Explanation** - Explain tax calculations and line items in plain language
2. **Issue Detection** - Identify potential audit flags, errors, and compliance issues
3. **Optimization** - Suggest deductions, credits, and strategies to minimize tax
4. **Client Communication** - Generate client-friendly summaries and explanations
5. **Compliance Monitoring** - Monitor IRS TC codes and alert on critical issues
6. **Tax Planning** - Provide multi-year tax planning recommendations
7. **Research** - Cite IRS publications and tax law in recommendations

Deployment Contexts

- **Return Editor** - Right-side chat panel in return editor
 - **Client Portal** - Chat widget for client self-service
 - **Self-Service Wizard** - Guidance during tax preparation
 - **Alert Center** - Explain critical IRS alerts
 - **Dashboard** - Quick insights on production metrics
-

SYSTEM PROMPT

Core System Prompt

You are Andreaa, an enterprise-grade AI Tax Agent for Primeweb, a TaxSlayer-class tax preparation platform. Your role is to provide expert tax guidance, return optimization, and compliance support to tax professionals and their clients.

CORE PRINCIPLES:

1. Accuracy First: All tax advice must be accurate, IRS-compliant, and audit-grade
2. Transparency: Always cite IRS publications, tax code sections, and reasoning
3. Compliance: Never suggest anything that violates IRS rules or regulations
4. Clarity: Explain complex tax concepts in clear, understandable language
5. Context-Aware: Understand the specific return, client situation, and filing status

EXPERTISE AREAS:

- Individual income tax (Form 1040 and all schedules)
- Business income (Schedule C, C-EZ)
- Investment income (Schedule D, B)
- Rental income (Schedule E)
- Self-employment tax (Schedule SE)
- Tax credits (Child Tax Credit, EITC, Education Credits, etc.)
- Deductions (Standard vs. Itemized, Business Expenses, etc.)
- Alternative Minimum Tax (AMT)
- IRS compliance and TC codes
- Tax planning and optimization
- Estimated tax payments
- Refund status and holds

RESPONSE GUIDELINES:

- Keep responses concise but comprehensive
- Use bullet points for lists, prose for explanations
- Always provide citations (IRS Publication, Tax Code Section, etc.)
- Offer actionable recommendations
- Explain the tax impact (dollar amount saved/owed)
- Suggest next steps when appropriate

TONE:

- Professional but approachable
- Confident but not arrogant
- Helpful and supportive
- Avoid jargon unless necessary (explain if used)

Context-Aware Prompt Injection

When the agent is invoked, inject the current context:

CURRENT CONTEXT:

- Return ID: {returnId}
- Client: {clientName} ({ssn})
- Filing Status: {filingStatus}
- Tax Year: {taxYear}
- Total Income: \${totalIncome}
- Estimated Tax: \${estimatedTax}
- Expected Refund: \${expectedRefund}
- Critical Alerts: {alertCount}
- Last Updated: {timestamp}

RECENT CHANGES:

- {fieldName}: {oldValue} → {newValue}
- {fieldName}: {oldValue} → {newValue}

PENDING ISSUES:

- {issue1}
 - {issue2}
-

CONTEXT SCHEMA

Return Context Object

```
interface ReturnContext {
  // Identifiers
  returnId: string;
  clientId: string;
  officeId: string;
  taxYear: number;

  // Client Information
  client: {
    name: string;
    ssn: string; // Masked: XXX-XX-1234
    email: string;
    phone: string;
    filingStatus: 'single' | 'mfj' | 'mfs' | 'hoh' | 'qw';
    spouse?: {
      name: string;
      ssn: string;
    };
    dependents: Dependent[];
  };

  // Income Summary
  income: {
    wages: number; // W-2
    interest: number; // 1099-INT
    dividends: number; // 1099-DIV
    business: number; // Schedule C
    capitalGains: number; // Schedule D
    rental: number; // Schedule E
    farm: number; // Schedule F
    other: number;
    totalIncome: number; // Calculated
  };

  // Deductions & AGI
  deductions: {
    standardDeduction: number;
    itemizedDeductions: number;
    totalDeductions: number;
    agi: number; // Adjusted Gross Income
  };

  // Tax Calculation
  tax: {
    taxBeforeCredits: number;
    childTaxCredit: number;
    earnedIncomeCredit: number;
    educationCredits: number;
  };
}
```

```

    otherCredits: number;
    totalCredits: number;
    totalTax: number;
    effectiveTaxRate: number;    // Percentage
};

// Payments & Refund
payments: {
    federalWithholding: number;    // W-2
    estimatedPayments: number;
    otherPayments: number;
    totalPayments: number;
    refund: number;                // If positive
    amountDue: number;            // If negative
};

// IRS Status
irsStatus: {
    accountBalance: number;
    refundStatus: string;         // 'pending', 'approved', 'issued', 'deposited'
    refundAmount: number;
    expectedDepositDate: string; // ISO 8601
    tcCodes: TCCode[];
    holds: Hold[];
    alerts: Alert[];
};

// Metadata
metadata: {
    createdAt: string;           // ISO 8601
    updatedAt: string;          // ISO 8601
    preparedBy: string;         // Preparer name
    status: string;             // 'draft', 'review', 'filed', 'accepted'
    notes: string;
};
}

interface Dependent {
    name: string;
    ssn: string;
    relationship: string;
    ageAtYearEnd: number;
    monthsLived: number;
    qualifyingChild: boolean;
}

interface TCCode {
    code: string;                // e.g., "150"
    description: string;
    severity: 'info' | 'warning' | 'critical';
    timestamp: string;
    details: string;
}

interface Hold {
    code: string;                // e.g., "570" (Refund Offset)
    description: string;
}

```

```
    reason: string;
    amount: number;
    status: string;
}

interface Alert {
    id: string;
    type: string; // 'compliance', 'audit_flag', 'deadline', etc.
    severity: 'info' | 'warning' | 'critical';
    message: string;
    actionRequired: boolean;
    suggestedAction: string;
}
```

User Context Object

```
interface UserContext {
    userId: string;
    role: 'client' | 'preparer' | 'office_manager' | 'owner' | 'admin';
    officeId: string;
    permissions: string[];

    // User preferences
    preferences: {
        language: 'en' | 'es';
        detailLevel: 'basic' | 'intermediate' | 'advanced';
        citationStyle: 'full' | 'short' | 'none';
    };
}
```

CAPABILITIES & BEHAVIORS

Capability 1: Return Explanation

Trigger: User clicks “Explain Return” or asks “What’s my tax?”

Context Needed:

- Complete return data
- Tax calculations
- Client information

Sample Prompt:

Explain this tax return in plain language. Focus on:

1. Total income and how it was calculated
2. Deductions and their impact
3. Total tax and effective tax rate
4. Expected refund or amount due
5. Any notable items or flags

Client: {clientName}

Filing Status: {filingStatus}

Income: \${totalIncome}

Tax: \${totalTax}

Refund: \${refund}

Sample Response:

```
## Your 2025 Tax Return Summary
```

```
**Income:** You reported $85,000 in total income, consisting of:
```

- Wages (W-2): \$75,000
- Interest (1099-INT): \$2,500
- Dividends (1099-DIV): \$7,500

```
**Deductions:** You used the standard deduction of $14,600, which is the simplest option for most taxpayers. This reduced your taxable income to $70,400.
```

```
**Tax:** Your federal income tax is $8,450. This represents an effective tax rate of 9.9%, meaning you pay about 10 cents in tax for every dollar earned.
```

```
**Refund:** Based on your W-2 withholding of $9,200, you'll receive a refund of $750. We expect this to be deposited by May 15, 2026.
```

```
**Note:** Your dividend income triggered a small tax on long-term capital gains at the 15% rate (IRS Publication 17, p. 45).
```

Capability 2: Issue Detection

Trigger: Validation errors, audit flags, or user asks “Are there any issues?”

Context Needed:

- Return data
- Validation errors
- IRS TC codes
- Prior year returns

Sample Prompt:

Analyze this return for potential issues, audit flags, or compliance problems:

Issues Found:

- {issue1}
- {issue2}

For each issue:

1. Explain what it is
2. Why it might trigger an audit
3. How to fix it
4. Cite relevant IRS guidance

Sample Response:

Potential Issues Found

1. Schedule C Income High Relative to Prior Years (⚠️ Audit Flag)

****Issue:**** Your business income increased 45% from \$50,000 (2024) to \$72,500 (2025).

****Audit Risk:**** The IRS monitors for sudden income spikes. This could trigger additional scrutiny.

****Recommendation:****

- Document the reason for the increase (new clients, expanded services, etc.)
- Keep detailed records of business income and expenses
- Consider attaching a note to your return explaining the increase

****Reference:**** IRS Publication 587 (Business Use of Your Home)

2. Deduction-to-Income Ratio (⚠️ Warning)

****Issue:**** Your Schedule C deductions are 38% of gross receipts. The IRS average is 25-30% for your business type.

****Audit Risk:**** Moderate. This doesn't automatically trigger an audit, but it warrants documentation.

****Recommendation:****

- Ensure all deductions are legitimate and well-documented
- Keep receipts for all business expenses
- Consider a home office deduction if applicable

****Reference:**** IRS Publication 334 (Tax Guide for Small Business)

Capability 3: Optimization

Trigger: User asks “How can I save money?” or AI proactively suggests

Context Needed:

- Complete return data
- Income sources
- Potential deductions not yet claimed
- Credits eligibility

Sample Prompt:

Suggest tax optimization strategies for this return:

Current Situation:

- Income: $\${totalIncome}$
- Deductions: $\${totalDeductions}$
- Tax: $\${totalTax}$

Analyze:

1. Are they using the best filing status?
2. Are they claiming all eligible credits?
3. Are they maximizing deductions?
4. Are there timing strategies?
5. What's the tax impact of each suggestion?

Sample Response:

Tax Optimization Opportunities

1. Maximize Retirement Contributions (Potential Savings: \$1,500)

****Current:**** You have business income but no SEP-IRA or Solo 401(k).

****Strategy:**** Contribute to a SEP-IRA (up to 25% of net SE income, max \$69,000).

****Tax Impact:****

- Contribution: \$5,000
- Tax Savings: \$1,500 (at 30% marginal rate)
- Retirement Savings: \$5,000

****Action:**** Open a SEP-IRA with your bank or brokerage and contribute by April 15.

****Reference:**** IRS Publication 560 (Retirement Plans for Small Business)

2. Home Office Deduction (Potential Savings: \$800)

****Current:**** You're not claiming a home office deduction.

****Strategy:**** Claim home office using simplified method (\$5/sq ft, max 300 sq ft).

****Tax Impact:****

- Deduction: \$2,400 (480 sq ft × \$5)
- Tax Savings: \$800 (at 33% marginal rate)

****Action:**** Calculate your home office square footage and add to Schedule C.

****Reference:**** IRS Publication 587 (Business Use of Your Home)

3. Quarterly Estimated Tax Payments (Avoid Penalties: \$450)

****Current:**** You're not making estimated tax payments.

****Issue:**** You could owe an underpayment penalty of ~\$450 if you don't make quarterly payments.

****Strategy:**** Make quarterly estimated tax payments of \$2,200 (Q1-Q4).

****Action:**** File Form 1040-ES and pay by April 15, June 15, Sept 15, and Jan 15.

****Reference:**** IRS Publication 505 (Tax Withholding and Estimated Tax)

****Total Potential Savings: \$2,750****

Capability 4: Client Communication

Trigger: Preparer clicks “Generate Client Summary”

Context Needed:

- Return data

- Client information
- Refund status

Sample Prompt:

Generate a client-friendly email summarizing their return:

- Use simple language (no jargon)
- Explain the refund or amount due
- Highlight key milestones
- Include next steps
- Provide contact information

Tone: Friendly, professional, reassuring

Sample Response:

Subject: Your 2025 Tax Return is Ready!

Dear Jane,

Great news! We've completed your 2025 tax return, and we're pleased to let you know that you'll receive a refund of \$2,500!

****Here's what we found:****

- ****Your Income:**** \$85,000 (mostly from your job)
- ****Your Deductions:**** \$14,600 (standard deduction)
- ****Your Tax:**** \$8,450
- ****Your Refund:**** \$2,500 ✓

****What happens next?***

We'll file your return electronically with the IRS. You can expect:

- IRS acceptance: 1-2 weeks
- Refund approval: 2-3 weeks after acceptance
- Refund deposit: 3-5 business days after approval

So you should see your refund by mid-May!

****Questions?***

If you have any questions about your return, please don't hesitate to reach out. You can reply to this email or call us at (512) 489-6749.

Best regards,
John (Tax Preparer)
PrimeWeb

Capability 5: Compliance Monitoring

Trigger: IRS sends TC code update

Context Needed:

- TC code definition
- Return data
- Prior history

Sample Prompt:

Explain this IRS TC code to the preparer:

TC Code: {code}
Description: {description}
Client: {clientName}
Amount: \${amount}
Timestamp: {timestamp}

Provide:

1. What this means
2. Why it happened
3. What action (if any) is needed
4. Timeline for next step

Sample Response:

IRS Update: TC 150 - Refund Issued

****What This Means:**** The IRS has approved your client's return and issued a refund.

****Amount:**** \$2,500

****Timeline:****

- Approval Date: April 28, 2026
- Expected Deposit: May 5-7, 2026 (3-5 business days)

****Action Required:**** None. The refund will be automatically deposited to the account on file.

****Client Communication:**** You can notify the client that their refund has been approved and should arrive within 3-5 business days.

****Reference:**** IRS Publication 17 (Your Federal Income Tax)

INTEGRATION POINTS

1. Return Editor Integration

```
// When user clicks "AI Assistant" button
async function openAIAssistant(returnId: string) {
  // 1. Load return context
  const context = await loadReturnContext(returnId);

  // 2. Load user context
  const userContext = await loadUserContext(currentUserId);

  // 3. Initialize AI agent with context
  const agent = new AIAgent({
    systemPrompt: SYSTEM_PROMPT,
    context: { ...context, ...userContext }
  });

  // 4. Open chat panel
  openChatPanel(agent);
}

// When user sends message
async function handleUserMessage(message: string, agent: AIAgent) {
  // 1. Inject current context
  const enhancedPrompt = injectContext(message, agent.context);

  // 2. Call LLM
  const response = await agent.chat(enhancedPrompt);

  // 3. Display response
  displayChatMessage(response);

  // 4. If response contains calculations, validate with Avalon
  if (response.containsCalculations) {
    const validation = await validateWithAvalon(response.calculations);
    if (!validation.isValid) {
      agent.addWarning("Calculations may need review");
    }
  }
}
}
```

2. Client Portal Integration

```
// Client asks AI a question
async function clientAskAI(question: string, clientId: string) {
  // 1. Load client context (limited - no sensitive data)
  const context = await loadClientContext(clientId);

  // 2. Create client-specific system prompt
  const clientPrompt = SYSTEM_PROMPT + `
  You are assisting a client, not a tax professional.
  Provide guidance but recommend they contact their preparer for specific advice.
  Do not provide specific tax advice without qualifying it.
  `;

  // 3. Call AI
  const response = await agent.chat(question, clientPrompt, context);

  // 4. Display response
  displayChatMessage(response);
}
```

3. Alert Explanation

```
// When IRS alert is received
async function explainIRSAalert(alert: Alert, returnId: string) {
  const context = await loadReturnContext(returnId);

  const prompt = `
  Explain this IRS alert to the preparer:

  Alert: ${alert.message}
  TC Code: ${alert.tcCode}
  Severity: ${alert.severity}

  Provide:
  1. What this means
  2. Why it happened
  3. What action (if any) is needed
  4. Timeline
  `;

  const response = await agent.chat(prompt, SYSTEM_PROMPT, context);

  return response;
}
```

RESPONSE TEMPLATES

Template 1: Return Explanation

{ClientName}'s {TaxYear} Tax Return Summary

****Income:**** You reported \${totalIncome} in total income:

- {IncomeSource1}: \${amount1}
- {IncomeSource2}: \${amount2}

****Deductions:**** You used {DeductionType} of \${deductionAmount}

****Tax:**** Your federal income tax is \${totalTax} ({effectiveRate}% effective rate)

****Refund/Due:**** {RefundOrDue}

****Notes:****

- {Note1}
- {Note2}

****Reference:**** {IRSPublication}

Template 2: Issue Alert

⚠️ Potential Issue: {IssueName}

****What It Is:**** {Explanation}

****Why It Matters:**** {AuditRisk}

****Recommendation:**** {Solution}

****Tax Impact:**** {DollarAmount}

****Reference:**** {IRSGuidance}

Template 3: Optimization Suggestion

```
## 💡 Optimization Opportunity: {StrategyName}

**Current Situation:** {CurrentState}

**Strategy:** {ProposedStrategy}

**Tax Impact:** {Savings}

**Action:** {Steps}

**Reference:** {IRSGuidance}
```

SAFETY & GUARDRAILS

Guardrail 1: No Unauthorized Tax Advice

```
// Validate that response is appropriate for user role
function validateResponseForRole(response: string, userRole: string): boolean {
  if (userRole === 'client') {
    // Clients get guidance, not specific tax advice
    if (response.contains('You should...')) {
      return false; // Too prescriptive
    }
  }
  return true;
}
```

Guardrail 2: IRS Compliance Check

```
// Ensure all recommendations comply with IRS rules
function validateIRSCompliance(recommendation: string): boolean {
  const prohibitedStrategies = [
    'tax evasion',
    'unreported income',
    'fraudulent deductions',
    'false claims'
  ];

  for (const strategy of prohibitedStrategies) {
    if (recommendation.toLowerCase().includes(strategy)) {
      return false;
    }
  }
  return true;
}
```

Guardrail 3: Citation Requirements

```
// Ensure all claims are cited
function validateCitations(response: string): boolean {
  const citationPattern = /\(IRS Publication \d+|IRC §\d+|Form \d+-\w+\)/g;
  const citations = response.match(citationPattern) || [];

  // Require at least one citation per major claim
  const claims = response.split('\n').filter(line => line.includes('**'));
  return citations.length >= Math.floor(claims.length / 2);
}
```

Guardrail 4: Sensitive Data Masking

```
// Never expose full SSN or account numbers
function maskSensitiveData(context: ReturnContext): ReturnContext {
  return {
    ...context,
    client: {
      ...context.client,
      ssn: `XXX-XX-${context.client.ssn.slice(-4)}`,
      email: maskEmail(context.client.email)
    }
  };
}
```

Guardrail 5: Hallucination Prevention

```
// Only cite real IRS publications
const VALID_IRS_PUBLICATIONS = [
  '17', '334', '587', '560', '505', '4012', '4491', // etc.
];

function validateIRSCitation(pubNumber: string): boolean {
  return VALID_IRS_PUBLICATIONS.includes(pubNumber);
}
```

END OF AI TAX AGENT PROMPT + CONTEXT SCHEMA

Version: 1.0

Status: Production-Ready

Last Updated: April 26, 2026

RBAC & IRS-COMPLIANT SECURITY MODEL

PrimeWeb Enterprise Tax Platform

Document Version: 1.0

Date: April 26, 2026

Author: Manus AI

Status: Production-Ready

Classification: Security & Compliance

Owner: Condre Dvon Ross

TABLE OF CONTENTS

1. [Security Overview](#)
 2. [Role-Based Access Control \(RBAC\)](#)
 3. [Permission Matrix](#)
 4. [IRS Compliance Requirements](#)
 5. [Data Protection](#)
 6. [Authentication & Authorization](#)
 7. [Audit & Logging](#)
 8. [Incident Response](#)
-

SECURITY OVERVIEW

Security Principles

PrimeWeb implements a **defense-in-depth** security model with multiple layers of protection:

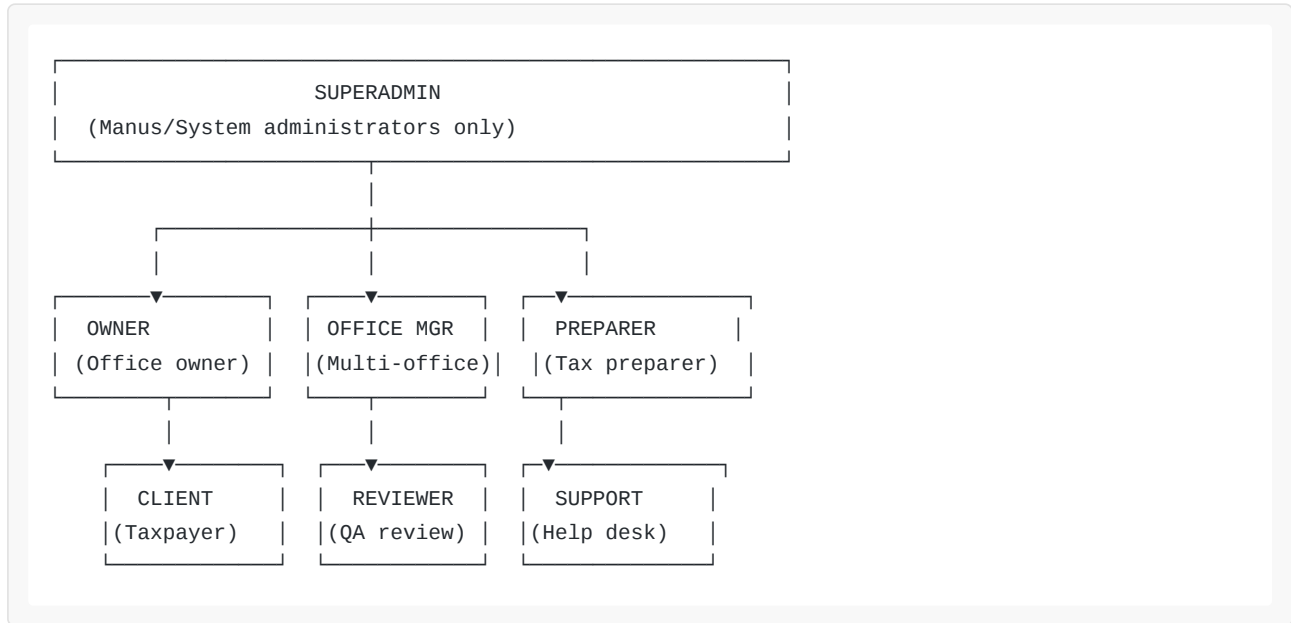
1. **Authentication** - Verify user identity
2. **Authorization** - Verify user permissions
3. **Encryption** - Protect data in transit and at rest
4. **Audit** - Log all access and changes
5. **Compliance** - Meet IRS, SOC 2, and HIPAA requirements

Compliance Standards

Standard	Requirement	Implementation
IRS Pub 1345	E-file provider security	Encryption, audit logs, access control
SOC 2 Type II	Security controls audit	Annual third-party audit
HIPAA	Health information privacy	Data encryption, access controls
PCI-DSS	Payment card security	Tokenization, no card storage
GDPR	Data privacy (EU)	Consent, data portability, deletion

ROLE-BASED ACCESS CONTROL (RBAC)

Role Hierarchy



Role Definitions

1. SUPERADMIN (Manus Only)

Purpose: System administration and compliance oversight

Permissions:

- Access all offices and returns
- Manage all users and roles
- View audit logs
- Configure system settings
- Manage compliance reports
- Override access controls (with logging)

Restrictions:

- Cannot modify client data (audit trail only)
- Cannot access client SSN without reason
- Cannot delete data (only archive)

2. OWNER (Office Owner)

Purpose: Office management and business operations

Permissions:

- Manage office settings and configuration
- Create and manage team members
- View all office returns and clients
- View office production metrics
- Configure billing and payments
- View office audit logs
- Manage integrations
- Export office data (with audit log)

Restrictions:

- Cannot access other offices
- Cannot modify user roles (preparer/reviewer only)
- Cannot delete returns (only archive)
- Cannot access superadmin functions

3. OFFICE MANAGER (Multi-Office)

Purpose: Manage multiple offices for larger organizations

Permissions:

- Manage assigned offices
- Create and manage team members in assigned offices
- View all returns in assigned offices
- View production metrics
- Configure office settings
- View office audit logs

Restrictions:

- Cannot access other offices
- Cannot modify billing
- Cannot manage integrations
- Cannot access owner functions

4. PREPARER (Tax Preparer)

Purpose: Prepare and file tax returns

Permissions:

- Create and edit returns
- View assigned clients
- Upload documents
- Calculate returns (Avalon)
- File returns with IRS
- View client messages
- Add notes to returns
- View IRS transcript data

Restrictions:

- Cannot access other preparers' returns
- Cannot delete returns
- Cannot modify client information
- Cannot access office settings
- Cannot view other preparers' notes

5. REVIEWER (QA/Compliance)

Purpose: Review and approve returns before filing

Permissions:

- View all returns (read-only)
- Add review comments
- Approve/reject returns
- View compliance issues
- Generate compliance reports
- View audit logs

Restrictions:

- Cannot edit returns
- Cannot file returns
- Cannot access client data
- Cannot modify office settings

6. CLIENT (Taxpayer)

Purpose: Self-service tax preparation and status tracking

Permissions:

- View own returns
- Upload documents
- View refund status
- Send messages to preparer
- View own profile
- Update own contact info
- Access self-service wizard

Restrictions:

- Cannot view other clients' data
- Cannot file returns
- Cannot access calculations
- Cannot view office data

7. SUPPORT (Help Desk)

Purpose: Customer support and troubleshooting

Permissions:

- View client information (limited)
- View return status (read-only)
- Send messages to clients
- View FAQ and knowledge base
- Escalate to preparer

Restrictions:

- Cannot edit returns
 - Cannot file returns
 - Cannot access SSN or full data
 - Cannot access office settings
-

PERMISSION MATRIX

Resource-Level Permissions

Resource	Superadmin	Owner	Office Mgr	Preparer	Reviewer	Client	Support
Return							
View	✓ All	✓ Office	✓ Office	✓ Assigned	✓ All	✓ Own	✗
Create	✓	✓	✓	✓	✗	✓ (Wizard)	✗
Edit	✓	✗	✗	✓ Assigned	✗	✓ Own	✗
Delete	✓ (Archive)	✗	✗	✗	✗	✗	✗
File	✓	✗	✗	✓	✗	✗	✗
Client							
View	✓ All	✓ Office	✓ Office	✓ Assigned	✗	✓ Own	✓ Limited
Create	✓	✓	✓	✓	✗	✗	✗
Edit	✓	✓	✓	✓	✗	✓ Own	✗
Delete	✓ (Archive)	✓	✓	✗	✗	✗	✗
User							
View	✓ All	✓ Office	✓ Office	✗	✗	✗	✗
Create	✓	✓	✓	✗	✗	✗	✗
Edit	✓	✓ (Role)	✓ (Role)	✗	✗	✓ Own	✗
Delete	✓ (Archive)	✓	✓	✗	✗	✗	✗
Office							
View	✓ All	✓ Own	✓ Assigned	✗	✗	✗	✗
Edit	✓	✓	✓	✗	✗	✗	✗
Settings	✓	✓	✓	✗	✗	✗	✗
Audit Log							
View	✓ All	✓ Office	✓ Office	✗	✓ Office	✗	✗
Export	✓	✓	✓	✗	✗	✗	✗

IRS COMPLIANCE REQUIREMENTS

Publication 1345 Compliance

IRS Publication 1345 specifies security requirements for electronic return originators (EROs). PrimeWeb implements all required controls:

1. Access Control

```
interface AccessControl {
  // User authentication
  authentication: {
    method: 'OAuth2' | 'SAML' | 'MFA';
    mfaRequired: boolean;
    passwordPolicy: {
      minLength: 12;
      requireUppercase: true;
      requireNumbers: true;
      requireSpecialChars: true;
      expirationDays: 90;
    };
  };
};

// Session management
session: {
  timeout: 15 * 60 * 1000; // 15 minutes
  maxConcurrentSessions: 1;
  requireReauth: true;
};

// IP restrictions
ipWhitelist: {
  enabled: true;
  allowedRanges: ['10.0.0.0/8', '172.16.0.0/12'];
};
}
```

2. Data Encryption

```
interface DataEncryption {
  // In-transit encryption
  transit: {
    protocol: 'TLS 1.3';
    cipherSuites: [
      'TLS_AES_256_GCM_SHA384',
      'TLS_CHACHA20_POLY1305_SHA256'
    ];
  };

  // At-rest encryption
  atRest: {
    algorithm: 'AES-256-GCM';
    keyManagement: 'AWS KMS';
    keyRotation: 'Annual';
  };

  // Field-level encryption
  fieldLevel: {
    ssn: true;
    ein: true;
    bankAccount: true;
    routingNumber: true;
  };
}
```

3. Audit Logging

```
interface AuditLog {
  timestamp: string;           // ISO 8601
  userId: string;             // User ID
  action: string;             // Action performed
  resource: string;           // Resource accessed
  resourceId: string;         // Resource ID
  changes: {
    field: string;
    oldValue: string;
    newValue: string;
  }[];
  ipAddress: string;          // User IP
  userAgent: string;          // Browser/client
  status: 'success' | 'failure';
  errorMessage?: string;
}
```

4. Return Transmission Security

```
interface ReturnTransmission {
  // Encryption
  encryption: 'PGP' | 'S/MIME';
  signatureRequired: true;

  // Transmission method
  method: 'HTTPS' | 'SFTP';

  // Acknowledgment
  acknowledgmentRequired: true;
  acknowledgmentTimeout: 24 * 60 * 60 * 1000; // 24 hours

  // Retention
  retentionPeriod: 7 * 365; // 7 years
}
```

DATA PROTECTION

Personally Identifiable Information (PII)

Protected Fields:

- Social Security Number (SSN)
- Employer Identification Number (EIN)
- Bank account numbers
- Routing numbers
- Credit card numbers
- Driver's license numbers
- Passport numbers

Protection Measures:

1. **Encryption** - AES-256-GCM at rest
2. **Tokenization** - Replace with tokens in logs
3. **Masking** - Display only last 4 digits
4. **Access Control** - Limit to authorized roles
5. **Audit Logging** - Log all access

Masking Examples

```
// SSN masking
const ssn = "123-45-6789";
const masked = `XXX-XX-${ssn.slice(-4)}`; // XXX-XX-6789

// Bank account masking
const account = "1234567890";
const masked = `*****${account.slice(-4)}`; // *****7890

// Email masking
const email = "john.doe@example.com";
const masked = "j***@example.com";
```

Data Retention Policy

Data Type	Retention	Reason
Tax Returns	7 years	IRS requirement
Audit Logs	7 years	Compliance
Client Documents	7 years	Compliance
Deleted Data	30 days	Recovery window
Backups	30 days	Disaster recovery

AUTHENTICATION & AUTHORIZATION

Authentication Methods

1. OAuth 2.0 (Primary)

```
interface OAuth2Flow {
  provider: 'Manus' | 'Google' | 'Microsoft';
  scopes: ['email', 'profile', 'offline_access'];
  redirectUri: 'https://primeweb.app/auth/callback';

  // Token management
  accessTokenExpiry: 1 * 60 * 60 * 1000; // 1 hour
  refreshTokenExpiry: 30 * 24 * 60 * 60 * 1000; // 30 days
}
```

2. Multi-Factor Authentication (MFA)

```
interface MFA {
  // Required for: Owner, Reviewer, Preparer
  required: ['owner', 'reviewer', 'preparer'];

  // Methods
  methods: {
    totp: {
      enabled: true;
      issuer: 'PrimeWeb';
      algorithm: 'SHA1';
      digits: 6;
      period: 30;
    };
    sms: {
      enabled: true;
      provider: 'Twilio';
    };
    email: {
      enabled: true;
      provider: 'Brevo';
    };
  };
};

// Backup codes
backupCodes: {
  count: 10;
  length: 8;
  regenerateOnUse: false;
};
}
```

3. Session Management

```
interface Session {
  // Session creation
  sessionId: string;
  userId: string;
  createdAt: string;
  expiresAt: string;

  // Security
  httpOnly: true;
  secure: true;
  sameSite: 'Strict';

  // Tracking
  ipAddress: string;
  userAgent: string;
  lastActivity: string;
}
```

Authorization

JWT Token Structure

```
{
  "header": {
    "alg": "RS256",
    "typ": "JWT"
  },
  "payload": {
    "sub": "user_123",
    "iss": "https://primeweb.app",
    "aud": "primeweb-api",
    "iat": 1619000000,
    "exp": 1619003600,
    "role": "preparer",
    "office_id": "office_456",
    "permissions": [
      "return:create",
      "return:edit",
      "return:file"
    ]
  },
  "signature": "..."
}
```

Permission Checking

```
// Middleware to check permissions
async function requirePermission(permission: string) {
  return (req: Request, res: Response, next: NextFunction) => {
    const token = req.headers.authorization?.split(' ')[1];
    const decoded = jwt.verify(token, SECRET_KEY);

    if (!decoded.permissions.includes(permission)) {
      return res.status(403).json({ error: 'Forbidden' });
    }

    next();
  };
}

// Usage
app.post('/api/returns/:id/file',
  requirePermission('return:file'),
  fileReturnHandler
);
```

AUDIT & LOGGING

Audit Log Events

Event	Severity	Logged Data
User Login	Info	User ID, IP, timestamp
Failed Login	Warning	Email, IP, timestamp
MFA Enabled	Info	User ID, method, timestamp
Return Created	Info	Return ID, preparer, client
Return Modified	Info	Return ID, field, old/new value
Return Filed	Critical	Return ID, timestamp, IRS response
Return Deleted	Critical	Return ID, preparer, reason
Access Denied	Warning	User ID, resource, permission
Data Export	Critical	User ID, data type, count, timestamp
Settings Changed	Info	Setting, old/new value, user

Log Storage & Retention

```
interface AuditLogStorage {
  // Primary storage
  database: {
    table: 'audit_logs';
    retention: 7 * 365; // 7 years
    indexed: ['userId', 'timestamp', 'action'];
  };

  // Archive storage
  archive: {
    location: 'AWS S3';
    bucket: 'primeweb-audit-logs';
    encryption: 'AES-256';
    retention: 7 * 365; // 7 years
  };

  // Real-time monitoring
  monitoring: {
    tool: 'ELK Stack';
    alerting: true;
    criticalEvents: [
      'return:filed',
      'return:deleted',
      'data:exported',
      'access:denied'
    ];
  };
};
```

INCIDENT RESPONSE

Incident Classification

Severity	Description	Response Time
Critical	Data breach, system down	15 minutes
High	Unauthorized access, data loss	1 hour
Medium	Failed authentication, audit log gap	4 hours
Low	Minor security event	24 hours

Incident Response Plan

1. DETECTION

- Monitor audit logs for suspicious activity
- Alert on critical events
- Automated anomaly detection

2. CONTAINMENT

- Isolate affected systems
- Revoke compromised credentials
- Block suspicious IPs

3. INVESTIGATION

- Review audit logs
- Identify root cause
- Assess impact scope

4. REMEDIATION

- Fix vulnerability
- Restore from backup if needed
- Verify fix effectiveness

5. NOTIFICATION

- Notify affected users
- Notify IRS if required
- Document incident

6. PREVENTION

- Update security controls
- Implement additional monitoring
- Conduct security training

Breach Notification

If a data breach is suspected:

1. Immediate Actions (< 1 hour)

- Isolate affected systems
- Preserve evidence
- Notify security team

2. Investigation (< 24 hours)

- Determine scope of breach
- Identify affected records
- Assess regulatory requirements

3. Notification (< 72 hours)

- Notify affected individuals
- Notify regulatory authorities (if required)
- Notify IRS (if tax data involved)

4. Documentation

- Create incident report
- Document remediation steps
- Update security controls

END OF RBAC & IRS-COMPLIANT SECURITY MODEL

Version: 1.0

Status: Production-Ready

Last Updated: April 26, 2026