**WSC SPORTS TRAVEL EXPENSE MANAGEMENT SYSTEM**

**IMPLEMENTATION TODO LIST - ORGANIZED BY TABS**

**Version:** 1.0  
**Target Stack:** Web Application + Snowflake + SQLite/PostgreSQL  
**Organization:** By Navigation Tabs  
**Numbering:** Each task has a unique number for easy reference with Copilot

**📋 PHASE 0: FOUNDATION & SETUP**

**0.3 Core Dependencies**

* [ ] **Task 010** - Install UI framework and component library
* [ ] **Task 011** - Install date formatting library (moment.js/date-fns)
* [ ] **Task 012** - Install CSV parsing library (Papa Parse)
* [ ] **Task 013** - Install PDF generation library (jsPDF/PDFKit)
* [ ] **Task 014** - Install Excel export library (xlsx/ExcelJS)
* [ ] **Task 015** - Install HTTP client (Axios/Fetch)
* [ ] **Task 016** - Install state management (Redux/Vuex/Context API)

**0.4 Design System**

* [ ] **Task 017** - Set up fonts: Sora (headers), Inter Tight (body)
* [ ] **Task 018** - Create color palette and theme variables
* [ ] **Task 019** - Create base CSS with formatting standards
* [ ] **Task 020** - Create reusable UI components:
  + [ ] **Task 020a** - Button component
  + [ ] **Task 020b** - Input/Select component
  + [ ] **Task 020c** - Date picker component
  + [ ] **Task 020d** - Table component with features (sort/filter/resize/reorder)
  + [ ] **Task 020e** - Modal/Dialog component
  + [ ] **Task 020f** - Toast notification component
  + [ ] **Task 020g** - Filter panel component
  + [ ] **Task 020h** - Loading spinner component

**📊 PHASE 1: DATABASE SCHEMA & MODELS**

**1.1 Create Lookup Tables (with Add/Edit/Delete functionality)**

* [ ] **Task 021** - Create **Sources** table (Navan, Agent, Manual)
* [ ] **Task 022** - Create **Categories** table (with emoji: ✈️ Airfare, 🏨 Lodging, 🚗 Car, etc.)
* [ ] **Task 023** - Create **Purposes** table (with emoji: 💼 Business trip, 📲 Onboarding, etc.)
* [ ] **Task 024** - Create **CabinClasses** table (with emoji: 💺 Economy, 🛫 Premium economy, etc.)
* [ ] **Task 025** - Create **TripTypes** table (with emoji: 🏠 Domestic, 🌍 International, 📍 Local)
* [ ] **Task 026** - Create **TripStatus** table (with emoji: 🔴 Cancelled, ⚠️ Upcoming, etc.)
* [ ] **Task 027** - Create **ValidationStatus** table (⚠️ Not ready, 🟡 Ready, 🟢 Validated)
* [ ] **Task 028** - Create **BookingTypes** table (✈️ Flight, 🏨 Hotel, 🚗 Car, 🚂 Train)
* [ ] **Task 029** - Create **BookingStatus** table (🔴 Cancelled, 🟢 Approved)
* [ ] **Task 030** - Create **Owners** table (Maryan Chesler, Martina Popinsk)

**1.2 Create Main Tables**

* [ ] **Task 031** - Create **ConsolidatedTransactions** table with ALL fields from spec
  + Core identity fields (Source, Employee, UserID, Email, FirstName, etc.)
  + Transaction details (ID, dates, type, category, vendor, etc.)
  + Financial fields (Currency, Amount, ExchangeRate, AmountUSD)
  + Flags & validation (Owner, PTO flag, validation flags, audit fields)
* [ ] **Task 032** - Create **ConsolidatedBookings** table (backend only)
  + BookingID, dates, type, status, vendor
  + Origin/destination countries and cities
* [ ] **Task 033** - Create **Trips** table with ALL fields from spec
  + Core identity (TripID auto-generated, email, employee info)
  + Date & duration fields (StartDate, EndDate, PTO tracking, Duration, Nights)
  + Location fields (6 countries, cities, trip description - calculated)
  + Financial & classification (Cabin class, Purpose, Status, ValidationStatus)
  + Audit fields (Owner, AuditID, timestamps, modified by)
* [ ] **Task 034** - Create **Headcount** table (CSV import monthly)
  + Period, UserID, Email, FirstName, LastName
  + Subsidiary, Site, Department, Domain, CostCenter
* [ ] **Task 035** - Create **Tax** table (CSV import)
  + FiscalYear, Country, Subsidiary
  + MealsCap, LodgingCap, TaxShield
* [ ] **Task 036** - Create **CountriesAndCities** table (CSV import)
  + Country, City
  + Allow manual entry if not in list
* [ ] **Task 037** - Create **AuditLog** table
  + AuditID, Timestamp, UserID, Action
  + LinkedTable, LinkedRecordID
  + OldValue (JSON), NewValue (JSON), Comments

**1.3 Create Database Indexes**

* [ ] **Task 038** - Create index on Transactions: Email, TransactionDate, CategoryId, TripId
* [ ] **Task 039** - Create index on Trips: Email, StartDate, EndDate, StatusId
* [ ] **Task 040** - Create index on AuditLog: Timestamp, LinkedTable, LinkedRecordID
* [ ] **Task 041** - Create index on Headcount: Email, Period

**1.4 Create Entity Models/Classes**

* [ ] **Task 042** - Create Transaction model with all properties
* [ ] **Task 043** - Create Booking model
* [ ] **Task 044** - Create Trip model with calculated properties
* [ ] **Task 045** - Create Headcount model
* [ ] **Task 046** - Create Tax model
* [ ] **Task 047** - Create AuditLog model
* [ ] **Task 048** - Create all lookup models (Category, Purpose, Status, etc.)

**🏗️ PHASE 2: CORE SERVICES & UTILITIES**

**2.1 Data Access Layer**

* [ ] **Task 049** - Create DAO/Repository for Transactions
* [ ] **Task 050** - Create DAO/Repository for Bookings
* [ ] **Task 051** - Create DAO/Repository for Trips
* [ ] **Task 052** - Create DAO/Repository for Headcount
* [ ] **Task 053** - Create DAO/Repository for Tax
* [ ] **Task 054** - Create DAO/Repository for AuditLog
* [ ] **Task 055** - Create DAO/Repository for all lookup tables

**2.2 Business Logic Services**

* [ ] **Task 056** - Create **TransactionService**
  + CRUD operations
  + Link transaction to trip
  + Unlink transaction from trip
  + Split transaction logic
  + Validation checks
* [ ] **Task 057** - Create **TripService**
  + CRUD operations
  + Calculate duration/nights
  + Calculate trip description
  + Auto-derive status from dates
  + Validation checks (date overlaps, etc.)
* [ ] **Task 058** - Create **MatchingService**
  + Manual matching logic
  + Auto-matching suggestions algorithm
  + Confidence scoring
* [ ] **Task 059** - Create **SplitService**
  + Manual split logic
  + Auto-split suggestions algorithm
  + Participant validation
* [ ] **Task 060** - Create **TripSuggestionService**
  + Clustering algorithm (group by email + date proximity)
  + Merge gap logic
  + Local trip detection
  + Nearby unlinked transactions finder
* [ ] **Task 061** - Create **TaxCalculationService**
  + Join logic with tax table
  + Meals exposure calculation
  + Lodging exposure calculation
  + Airfare cabin class flagging

**2.3 Import/Export Services**

* [ ] **Task 062** - Create **CSVImportService**
  + Parse Navan CSV with transformation rules
  + Parse Agent Google Sheets
  + Parse Headcount CSV
  + Parse Tax CSV
  + Parse Countries/Cities CSV
  + Category mapping based on Policy field
  + Exchange rate conversion
* [ ] **Task 063** - Create **ExportService**
  + Export to CSV
  + Export to Excel with formatting
  + Export to PDF

**2.4 Utility Functions**

* [ ] **Task 064** - Create date formatting utility (dd/mm/yyyy, dd/mm/yyyy hh:mm:ss)
* [ ] **Task 065** - Create number formatting utility (1,000.00, $1,000.00)
* [ ] **Task 066** - Create text formatting utility (capitalize first letter)
* [ ] **Task 067** - Create validation utility (email, date ranges, amounts)
* [ ] **Task 068** - Create message template generator utility
* [ ] **Task 069** - Create PDF declaration generator utility

**2.5 Audit Service**

* [ ] **Task 070** - Implement logging for all CRUD operations
* [ ] **Task 071** - Capture old/new values as JSON
* [ ] **Task 072** - Store user and timestamp
* [ ] **Task 073** - Implement restore functionality (except for split transactions)

**📑 TAB 1: REPORTS**

**1.1 Transactions Report**

* [ ] **Task 074** - Create GET /api/transactions endpoint (with pagination, filtering, sorting)
* [ ] **Task 075** - Create GET /api/transactions/:id endpoint
* [ ] **Task 076** - Create POST /api/transactions endpoint (create)
* [ ] **Task 077** - Create PUT /api/transactions/:id endpoint (update)
* [ ] **Task 078** - Create DELETE /api/transactions/:id endpoint
* [ ] **Task 079** - Create POST /api/transactions/:id/link-trip endpoint
* [ ] **Task 080** - Create POST /api/transactions/:id/unlink-trip endpoint
* [ ] **Task 081** - Create POST /api/transactions/:id/split endpoint
* [ ] **Task 082** - Create transactions list page (frontend)
* [ ] **Task 083** - Implement transaction table with all schema columns
* [ ] **Task 084** - Add filters (Owner, Email, Date range, Category, etc.)
* [ ] **Task 085** - Implement inline editing for editable fields
* [ ] **Task 086** - Add document hyperlink with icon
* [ ] **Task 087** - Implement sort by any column
* [ ] **Task 088** - Implement column resize functionality
* [ ] **Task 089** - Implement column reorder (drag & drop)
* [ ] **Task 090** - Add "Save View" functionality
* [ ] **Task 091** - Add "Export to CSV/Excel" button
* [ ] **Task 092** - Highlight editable fields (different background color)
* [ ] **Task 093** - Create "Edit Transaction" modal
  + All fields editable except ID
  + Date pickers for dates
  + Dropdowns for categories/cabin class
  + Participant selector/input
  + Save/Cancel buttons
* [ ] **Task 094** - Create "Delete Transaction" confirmation dialog
  + Show warning if linked to trip
* [ ] **Task 095** - Create "Link to Trip" dialog
  + Opens manual matching interface
* [ ] **Task 096** - Create "Split Transaction" dialog
  + Opens manual split interface
* [ ] **Task 097** - Create Transaction Detail View (double-click modal)
  + Document preview (if available)
  + All transaction details
  + Action buttons (Edit, Delete, Link, Split, Generate Message)

**1.2 Trips Report**

* [ ] **Task 098** - Create GET /api/trips endpoint (with pagination, filtering, sorting)
* [ ] **Task 099** - Create GET /api/trips/:id endpoint
* [ ] **Task 100** - Create GET /api/trips/:id/transactions endpoint (linked transactions)
* [ ] **Task 101** - Create POST /api/trips endpoint (create manual trip)
* [ ] **Task 102** - Create PUT /api/trips/:id endpoint (update)
* [ ] **Task 103** - Create DELETE /api/trips/:id endpoint
* [ ] **Task 104** - Create POST /api/trips/:id/link-transactions endpoint
* [ ] **Task 105** - Create POST /api/trips/:id/unlink-transactions endpoint
* [ ] **Task 106** - Create trips list page (frontend)
* [ ] **Task 107** - Implement trips table with all columns:
  + Trip ID, Traveler info, Dates, Duration, Nights
  + Countries/Cities, Purpose, Status, Validation Status
  + Owner, Last Modified info
* [ ] **Task 108** - Add comprehensive filters:
  + Owner, Email, Subsidiary, Site, Department, Domain
  + Date range, Purpose, Status, Validation Status
  + Free text search
* [ ] **Task 109** - Implement sort by any column
* [ ] **Task 110** - Implement column resize/reorder
* [ ] **Task 111** - Add "Save View" functionality
* [ ] **Task 112** - Add "Export to CSV/Excel" button
* [ ] **Task 113** - Create "Edit Trip" modal
  + All fields editable
  + Date overlap validation
  + Country/city dropdowns (up to 6)
  + Purpose/Status dropdowns
  + PTO checkbox with days input
  + Owner dropdown
  + Comments text area
* [ ] **Task 114** - Create "Delete Trip" confirmation dialog
  + Show linked transactions count
  + Option to unlink or prevent delete
* [ ] **Task 115** - Create "View Linked Transactions" panel
  + Show all linked transactions in table
  + "Link More Transactions" button
  + "Unlink" button for each transaction
* [ ] **Task 116** - Create Trip Detail View (double-click modal)
  + Trip information section
  + Linked transactions section (table)
  + Tax calculation section (with breakdown)
  + Action buttons

**1.3 Create Manual Trip**

* [ ] **Task 117** - Create POST /api/trips/manual endpoint
* [ ] **Task 118** - Create GET /api/employees endpoint (for dropdown)
* [ ] **Task 119** - Implement date range validation
* [ ] **Task 120** - Implement overlapping trips check
* [ ] **Task 121** - Implement auto-generate unique Trip ID
* [ ] **Task 122** - Create Manual Trip Form Modal (frontend)
* [ ] **Task 123** - Build Traveler Selection section:
  + Employee dropdown (if employee) with autocomplete
  + Auto-populate: Name, Email, Site, Subsidiary, Cost Center, Dept, Domain
  + Manual entry fields (if external guest)
* [ ] **Task 124** - Build Trip Details section:
  + Start/End date pickers (with validation)
  + Flag and warning for overlapping trips
  + PTO checkbox + days input (validation: 1 to duration)
  + Country/City selectors (up to 6 countries, dynamic city dropdown)
  + Purpose dropdown
  + Status (auto-derived, allow override to Cancelled)
  + Comments text area
* [ ] **Task 125** - Build Ownership & Validation section:
  + Owner dropdown (auto-fetched from cost center)
  + "Mark as Validated" checkbox
* [ ] **Task 126** - Implement validation logic:
  + Start date ≤ End date
  + PTO days ≤ Duration
  + At least 1 country required
* [ ] **Task 127** - Add Save/Cancel buttons
* [ ] **Task 128** - Show success notification on save

**1.4 Trip Suggestions**

* [ ] **Task 129** - Create POST /api/trips/suggestions endpoint (with filters)
* [ ] **Task 130** - Create GET /api/transactions/unlinked endpoint (for nearby search)
* [ ] **Task 131** - Create POST /api/trips/suggestions/create endpoint (bulk create trips)
* [ ] **Task 132** - Implement clustering algorithm:
  + Group by email
  + Build date intervals from booking dates
  + Sort and merge intervals (gap\_days parameter)
  + Attach transactions within date windows
  + Handle source trip ID grouping
  + Detect local trips (same-day, no travel booking)
  + Merge local trips if inside regular suggestion dates
* [ ] **Task 133** - Calculate suggestion metadata:
  + Transaction count, Total amount, Destinations (distinct countries/cities)
  + Day/night counts
* [ ] **Task 134** - Pull headcount snapshot as of start date
* [ ] **Task 135** - Create Trip Suggestions Page (frontend)
* [ ] **Task 136** - Build filters section:
  + Owner (multi-select)
  + Date range pickers
  + Merge gap slider (days)
* [ ] **Task 137** - Add action buttons: ✔️ Run (fetch suggestions), ❌ Clear filters
* [ ] **Task 138** - Create suggestions table:
  + Columns: Type (✈️/🏠), Email, Employee info, Dates, Duration, Nights, Destinations, Transaction count, Selected summary
  + Row selection checkboxes
* [ ] **Task 139** - Implement split panel on row selection:
  + **Left Panel: Create Trip Form**
    - Employee snapshot (non-editable display)
    - Start/End dates (editable)
    - Countries/Cities (editable)
    - Purpose dropdown, Owner dropdown (pre-filled, editable), Comments
  + **Right Panel: Transactions**
    - Editable grid of matched transactions
    - Include checkboxes for each
    - "Nearby Unlinked" expander (Input for ±N days, Auto-load unlinked transactions, Checkboxes to add them)
* [ ] **Task 140** - Add "Create Trip" button functionality:
  + Creates trip record
  + Links selected transactions
  + Updates UI
  + Shows success notification

**1.5 Trip Validation**

* [ ] **Task 141** - Create GET /api/trips/validation endpoint (filtered trips with Ready to Validate status)
* [ ] **Task 142** - Create PUT /api/trips/:id/validate endpoint (mark as validated)
* [ ] **Task 143** - Create GET /api/trips/:id/spend-breakdown endpoint
* [ ] **Task 144** - Implement validation flag calculations:
  + No airfare on non-local trip flag
  + No lodging on non-local trip flag
  + Non-economy cabin class flag
  + Lodging per night > $220 flag
  + Meals per day > $80 flag
  + Transportation per day > $100 flag
* [ ] **Task 145** - Create Trip Validation Page (frontend)
* [ ] **Task 146** - Build filters section:
  + Owner, Traveler, Department, Trip Purpose, Status, Site, Domain
  + Validation Status (default: Pending)
* [ ] **Task 147** - Create validation table:
  + Columns: Validation status, Traveler, Email, Site, Start/End dates, Duration, Destinations, Trip status, Purpose, # Transactions, Total amount
  + Row selection (radio button)
* [ ] **Task 148** - Create split view on selection:
  + **Left Panel: Trip Details**
    - Display all trip info, Editable fields
    - Show validation flags with icons/colors
    - Key metrics bullets: Total trip cost, Date range, Key destinations, Airfare cabin classes (distinct), Lodging per night, Meals per day, Transportation per day
    - "Validate" button at bottom
  + **Right Panel: Transactions**
    - Summary by category (ordered: Airfare, Lodging, Meals, Client entertainment, Transportation, Communication, Other)
    - Double-click to show full records (pivot table style)
    - Actions on transactions: Edit, Unlink, Link to another trip, Split, Delete

**1.6 Travel Spend Report**

* [ ] **Task 149** - Create GET /api/reports/travel-spend endpoint (aggregate data)
* [ ] **Task 150** - Calculate for each trip:
  + **of transactions, Total amount ($), Cost per day ($)**
  + Airfare ($) + distinct cabin classes
  + Lodging ($) + per night ($)
  + Meals ($) + per day ($)
  + Transportation ($) + per day ($)
  + Client Entertainment ($), Communication ($), Other ($)
  + Tax exposure (calculated)
* [ ] **Task 151** - Create Travel Spend Report page (frontend)
* [ ] **Task 152** - Implement report table with all trip columns + calculated spend columns
* [ ] **Task 153** - Add all standard filters (Owner, Email, Date range, etc.)
* [ ] **Task 154** - Implement sort/filter/resize/reorder
* [ ] **Task 155** - Add export functionality
* [ ] **Task 156** - Implement interactive features:
  + Double-click trip row → Show all linked transactions (modal/drawer)
  + Double-click tax exposure → Show calculation breakdown with caps and shields

**🔍 TAB 2: DATA INTEGRITY**

**2.1 Controls - Airfare**

* [ ] **Task 157** - Create GET /api/controls/airfare endpoint (WHERE Category = Airfare AND CabinClass IS NULL)
* [ ] **Task 158** - Create PUT /api/transactions/:id/cabin-class endpoint (update cabin class)
* [ ] **Task 159** - Create Airfare Control Page (frontend)
* [ ] **Task 160** - Build filtered table showing:
  + Document (hyperlink), Transaction ID, Email, Date, Vendor, Address
  + Currency, Amount, Amount (USD), Cabin Class (EDITABLE), Category (EDITABLE)
* [ ] **Task 161** - Highlight editable columns (Cabin Class, Category)
* [ ] **Task 162** - Implement inline editing
* [ ] **Task 163** - Set order by: Email, Date
* [ ] **Task 164** - Add filter by: Owner
* [ ] **Task 165** - Add bulk update capability

**2.2 Controls - Meals**

* [ ] **Task 166** - Create GET /api/controls/meals endpoint (WHERE Category = Meals AND ABS(AmountUSD) >= 80 AND NOT IsValid)
* [ ] **Task 167** - Create PUT /api/transactions/:id/mark-valid endpoint
* [ ] **Task 168** - Create POST /api/controls/meals/generate-message endpoint
* [ ] **Task 169** - Implement message template logic:
  + Detect participant types (external/internal/none)
  + Generate appropriate message template
  + Format for Slack (code blocks, inline code)
* [ ] **Task 170** - Create Meals Control Page (frontend)
* [ ] **Task 171** - Build filtered table showing:
  + Document, Transaction ID, Email, Date, Vendor, Address
  + Currency, Amount, Amount (USD), Participants, Category (EDITABLE)
* [ ] **Task 172** - Highlight Category column
* [ ] **Task 173** - Implement inline editing
* [ ] **Task 174** - Set order by: Email, Date
* [ ] **Task 175** - Add filters: Owner, Amount (USD) with default ≥$80
* [ ] **Task 176** - Add row actions: "Mark as Valid" button, "Generate Message" button (copies to clipboard)
* [ ] **Task 177** - Create message modal with three templates:
  + Template for external participants
  + Template for internal participants only
  + Template for no participants
* [ ] **Task 178** - Add "Copy to Clipboard" functionality

**2.3 Controls - Lodging**

* [ ] **Task 179** - Create GET /api/controls/lodging endpoint (WHERE Category = Lodging AND ABS(AmountUSD) <= 100 AND NOT IsValid)
* [ ] **Task 180** - Create PUT /api/transactions/:id/mark-valid endpoint
* [ ] **Task 181** - Create Lodging Control Page (frontend)
* [ ] **Task 182** - Build filtered table showing:
  + Document, Transaction ID, Email, Date, Vendor, Address
  + Currency, Amount, Amount (USD), Participants, Category (EDITABLE)
* [ ] **Task 183** - Highlight Category column
* [ ] **Task 184** - Implement inline editing
* [ ] **Task 185** - Set order by: Email, Date
* [ ] **Task 186** - Add filters: Owner, Amount (USD) with default ≤$100
* [ ] **Task 187** - Add "Mark as Valid" button for selected rows

**2.4 Controls - Client Entertainment**

* [ ] **Task 188** - Create GET /api/controls/client-entertainment endpoint (WHERE Category = Client entertainment AND NOT ParticipantsValidated)
* [ ] **Task 189** - Create PUT /api/transactions/:id/participants endpoint
* [ ] **Task 190** - Create POST /api/controls/client-entertainment/generate-message endpoint
* [ ] **Task 191** - Create Client Entertainment Control Page (frontend)
* [ ] **Task 192** - Build filtered table showing:
  + Document, Transaction ID, Email, Date, Vendor, Address
  + Currency, Amount, Amount (USD), Participants (EDITABLE), Category (EDITABLE)
* [ ] **Task 193** - Highlight Participants and Category columns
* [ ] **Task 194** - Implement participant editor:
  + Dropdown for internal employees (from Headcount)
  + Free text input for external participants (email format)
  + Concatenate with commas
  + Set ParticipantsValidated = TRUE on save
* [ ] **Task 195** - Set order by: Email, Date
* [ ] **Task 196** - Add filter: Owner
* [ ] **Task 197** - Add "Generate Message" button
* [ ] **Task 198** - Create message template formatted for Slack

**2.5 Controls - Other**

* [ ] **Task 199** - Create GET /api/controls/other endpoint (WHERE Category = Other)
* [ ] **Task 200** - Create POST /api/controls/other/generate-message endpoint
* [ ] **Task 201** - Create Other Control Page (frontend)
* [ ] **Task 202** - Build filtered table showing:
  + Document, Transaction ID, Email, Date, Vendor, Address
  + Currency, Amount, Amount (USD), Category (EDITABLE)
* [ ] **Task 203** - Highlight Category column
* [ ] **Task 204** - Implement inline editing
* [ ] **Task 205** - Set order by: Email, Date
* [ ] **Task 206** - Add filter: Owner
* [ ] **Task 207** - Add "Generate Message" button per row
* [ ] **Task 208** - Create message template with trip details

**2.6 Controls - Missing Documentation**

* [ ] **Task 209** - Create GET /api/controls/missing-docs endpoint (WHERE Documentation IS NULL AND ABS(AmountUSD) > 25)
* [ ] **Task 210** - Create POST /api/controls/missing-docs/generate-declarations endpoint
* [ ] **Task 211** - Create POST /api/controls/missing-docs/generate-message endpoint
* [ ] **Task 212** - Create PUT /api/transactions/:id/upload-document endpoint
* [ ] **Task 213** - Create PDF declaration generator:
  + PDF template for lost receipt declaration
  + Include fields: Transaction ID, Name, Amount, Date, Category, Vendor, Address
  + Add signature line
  + Generate one PDF per transaction or grouped by employee
* [ ] **Task 214** - Create Missing Documentation Page (frontend)
* [ ] **Task 215** - Build table grouped by email:
  + Document (placeholder/upload), Transaction ID, Email, Date, Category, Vendor, Address
  + Currency, Amount, Amount (USD)
* [ ] **Task 216** - Add group-level actions:
  + "Generate Declarations" (download PDFs)
  + "Generate Message Template" (copy to clipboard)
* [ ] **Task 217** - Add row-level action:
  + "Upload Document" (file upload, updates record)
* [ ] **Task 218** - Create Slack-formatted message template:
  + List all transactions with code blocks
  + Friendly request for receipts
  + Mention declaration forms for lost receipts

**2.7 Matching Engine - Manual Matching**

* [ ] **Task 219** - Create GET /api/trips/for-matching endpoint (with email filter)
* [ ] **Task 220** - Create GET /api/transactions/unmatched endpoint (with email filter)
* [ ] **Task 221** - Create POST /api/matching/manual endpoint (link transactions to trip)
* [ ] **Task 222** - Create Manual Matching Page (frontend)
* [ ] **Task 223** - Build top section:
  + Email dropdown/multi-select (with "All owners" option)
  + Checkbox: "Show matched transactions as well"
* [ ] **Task 224** - Create split screen layout:
  + **Left Screen: Trips**
    - Table with columns: Trip ID, Identifier, Start Date, End Date, Duration, Purpose, Status
    - Radio button selection (single trip)
    - Toggle to show already-matched (with icon indicator)
  + **Right Screen: Unmatched Transactions**
    - Table with columns: Document link, Transaction ID, Date, Booking dates (dd/mm - dd/mm), Category, Vendor, Address, Currency, Amount, Amount (USD)
    - Checkbox selection (multiple transactions)
* [ ] **Task 225** - Add "Link Selected" button
* [ ] **Task 226** - Implement linking logic:
  + Update selected transactions with Trip ID
  + Refresh both tables
  + Show success notification

**2.8 Matching Engine - Matching Suggestions**

* [ ] **Task 227** - Create POST /api/matching/suggestions endpoint (with filters)
* [ ] **Task 228** - Create POST /api/matching/accept-suggestions endpoint (bulk link)
* [ ] **Task 229** - Implement auto-matching logic:
  + Matching criteria: Booking start & end within trip dates
  + Transaction or authorization date within trip window
  + Source trip ID matches existing trip ID for traveler
  + Run only on transactions where Trip ID is null
  + Calculate confidence level/reason
* [ ] **Task 230** - Create Matching Suggestions Page (frontend)
* [ ] **Task 231** - Build filters section:
  + Owner (dropdown), Email (dropdown), Search button
* [ ] **Task 232** - Create suggestions table:
  + Checkbox, Document link, Transaction ID, Email, Date, Category, Vendor, Address
  + Currency, Amount, Amount (USD), Trip Description, Reason
* [ ] **Task 233** - Add row selection (checkboxes)
* [ ] **Task 234** - Add "Accept Selected Matches" button
* [ ] **Task 235** - Add "Reject" action per row
* [ ] **Task 236** - Show confidence indicators (color/icon)

**2.9 Split Engine - Split Suggestions**

* [ ] **Task 237** - Create POST /api/split/suggestions endpoint (with filters)
* [ ] **Task 238** - Create GET /api/transactions/:id/split-details endpoint
* [ ] **Task 239** - Implement auto-split logic:
  + Split criteria: Lodging >$150 with multiple WSC employees
  + Client entertainment >$180 with multiple WSC employees
  + Meals >$80 with multiple WSC employees
  + Validate participants exist in headcount table
  + Calculate confidence level (High/Medium/Low/Manual Review Required)
* [ ] **Task 240** - Create Split Suggestions Page (frontend)
* [ ] **Task 241** - Build filters section:
  + Owner (dropdown), Category (dropdown), Email (dropdown), Confidence Level (dropdown)
  + Checkboxes: "Exclude missing documentation", "Exclude unvalidated participants"
  + Search button
* [ ] **Task 242** - Show active filters bar below
* [ ] **Task 243** - Create expandable row list:
  + **Collapsed view**: Category icon, Merchant name, Transaction date, Amount, Status dot (employees in headcount?)
* [ ] **Task 244** - Implement expanded view:
  + Split/Discard buttons
  + **Transaction Details section**: Receipt link/attachment, Total amount and per-person split, Transaction date, Category, Merchant name
  + **Participants section**: List of participant emails, Validation icons (red if not found in headcount)
* [ ] **Task 245** - Add tooltips for checkbox filters

**2.10 Split Engine - Manual Split**

* [ ] **Task 246** - Create POST /api/transactions/:id/split endpoint
* [ ] **Task 247** - Create GET /api/trips/suggest-for-participant endpoint (auto-suggest trip)
* [ ] **Task 248** - Implement split logic:
  + Update original transaction (New amount and amount USD, Remove split participants, Set split flag = false)
  + For each split record: Generate unique ID, Copy all values except employee-specific fields, Fetch from headcount table if employee, Update participants field, Set split flag = false, Update trip ID if relevant
  + Log in audit trail
* [ ] **Task 249** - Create Manual Split Modal (frontend)
* [ ] **Task 250** - Display transaction info: Transaction ID, Original Amount
* [ ] **Task 251** - Create participants section:
  + Multi-select employee picker
  + Add external participant input (email)
  + Display participant badges with remove option
* [ ] **Task 252** - Calculate and show split details:
  + Amount per participant, Visual breakdown
* [ ] **Task 253** - Add trip selection per participant:
  + Dropdown with suggested trips, "❌ No Trip" option
* [ ] **Task 254** - Show summary section: Final split amounts, Participant details
* [ ] **Task 255** - Add action buttons: ▶️ Execute Split, ❌ Cancel
* [ ] **Task 256** - Validate before execution:
  + At least 2 participants, Amounts equal original total
* [ ] **Task 257** - Show success notification after split

**⚙️ TAB 3: SETTINGS**

**3.1 Settings - Lists Manager**

* [ ] **Task 258** - Create GET /api/settings/lists/:listType endpoint (get all items)
* [ ] **Task 259** - Create POST /api/settings/lists/:listType endpoint (add item)
* [ ] **Task 260** - Create PUT /api/settings/lists/:listType/:id endpoint (update item)
* [ ] **Task 261** - Create DELETE /api/settings/lists/:listType/:id endpoint (delete item)
* [ ] **Task 262** - Create Lists Manager Page (frontend)
* [ ] **Task 263** - Build dropdown to select list type:
  + Sources, Categories, Purposes, Cabin Classes, Trip Types, Trip Status, Validation Status, Booking Types, Booking Status, Owners
* [ ] **Task 264** - Create table for selected list:
  + Columns: ID, Emoji (if applicable), Name
  + Actions: Edit, Delete buttons per row
* [ ] **Task 265** - Add "Add New Item" button
* [ ] **Task 266** - Create Add/Edit modal:
  + Emoji picker (if applicable)
  + Name input
  + Save/Cancel buttons
* [ ] **Task 267** - Add confirmation dialog for delete
* [ ] **Task 268** - Show success/error notifications

**3.2 Settings - Data Transformation (Quick Rules)**

* [ ] **Task 269** - Create GET /api/settings/transformation-rules endpoint
* [ ] **Task 270** - Create POST /api/settings/transformation-rules endpoint (add rule)
* [ ] **Task 271** - Create PUT /api/settings/transformation-rules/:id endpoint (update rule)
* [ ] **Task 272** - Create DELETE /api/settings/transformation-rules/:id endpoint (delete rule)
* [ ] **Task 273** - Implement category mapping rules storage:
  + Policy field contains → Category mapping
  + Apply during CSV import
  + Examples from spec (tripactions\_fees → Trip fee, Airalo → Communication, etc.)
* [ ] **Task 274** - Create Data Transformation Page (frontend)
* [ ] **Task 275** - Build rules table:
  + Columns: Policy Contains, Arrow (→), Category, Actions
* [ ] **Task 276** - Add "Add New Rule" button
* [ ] **Task 277** - Create Add/Edit modal:
  + Policy text input (contains)
  + Category dropdown
  + Save/Cancel buttons
* [ ] **Task 278** - Add drag-to-reorder functionality (priority)
* [ ] **Task 279** - Add delete confirmation
* [ ] **Task 280** - Add import/export rules functionality

**3.3 Settings - Countries and Cities**

* [ ] **Task 281** - Create GET /api/settings/countries-cities endpoint
* [ ] **Task 282** - Create POST /api/settings/countries-cities/import endpoint (CSV upload)
* [ ] **Task 283** - Create POST /api/settings/countries-cities endpoint (add manual entry)
* [ ] **Task 284** - Create PUT /api/settings/countries-cities/:id endpoint (update)
* [ ] **Task 285** - Create DELETE /api/settings/countries-cities/:id endpoint (delete)
* [ ] **Task 286** - Create Countries and Cities Page (frontend)
* [ ] **Task 287** - Build import section:
  + File upload for CSV
  + Import button
  + Progress indicator
* [ ] **Task 288** - Create table:
  + Columns: Country, City, Source (imported/manual), Actions
* [ ] **Task 289** - Add filters: Country, City search
* [ ] **Task 290** - Add "Add Manual Entry" button
* [ ] **Task 291** - Create Add/Edit modal:
  + Country input/dropdown
  + City input
  + Save/Cancel buttons
* [ ] **Task 292** - Add export functionality

**3.4 Settings - Tax Settings**

* [ ] **Task 293** - Create GET /api/settings/tax endpoint
* [ ] **Task 294** - Create POST /api/settings/tax/import endpoint (CSV upload)
* [ ] **Task 295** - Create POST /api/settings/tax endpoint (add manual entry)
* [ ] **Task 296** - Create PUT /api/settings/tax/:id endpoint (update)
* [ ] **Task 297** - Create DELETE /api/settings/tax/:id endpoint (delete)
* [ ] **Task 298** - Create Tax Settings Page (frontend)
* [ ] **Task 299** - Build import section:
  + File upload for Tax CSV
  + Import button
  + Progress indicator
* [ ] **Task 300** - Create table:
  + Columns: Fiscal Year, Country, Subsidiary, Meals Cap, Lodging Cap, Tax Shield, Actions
* [ ] **Task 301** - Add filters: Fiscal Year, Country, Subsidiary
* [ ] **Task 302** - Add "Add Manual Entry" button
* [ ] **Task 303** - Create Add/Edit modal:
  + Fiscal Year input (4-digit)
  + Country dropdown
  + Subsidiary input
  + Meals Cap, Lodging Cap, Tax Shield inputs (numeric with validation)
  + Save/Cancel buttons
* [ ] **Task 304** - Add validation: All cap amounts must be positive numbers
* [ ] **Task 305** - Add export functionality

**3.5 Settings - Owners Management**

* [ ] **Task 306** - Create GET /api/settings/owners endpoint
* [ ] **Task 307** - Create POST /api/settings/owners endpoint (add owner)
* [ ] **Task 308** - Create PUT /api/settings/owners/:id endpoint (update)
* [ ] **Task 309** - Create DELETE /api/settings/owners/:id endpoint (delete)
* [ ] **Task 310** - Create Owners Management Page (frontend)
* [ ] **Task 311** - Build table:
  + Columns: Cost Center, Department, Domain, Owner(s), Actions
* [ ] **Task 312** - Add "Add New Owner" button
* [ ] **Task 313** - Create Add/Edit modal:
  + Cost Center input
  + Department input
  + Domain input
  + Owner multi-select/dropdown
  + Save/Cancel buttons
* [ ] **Task 314** - Add delete confirmation

**3.6 Audit Log**

* [ ] **Task 315** - Create GET /api/audit-log endpoint (with pagination and filters)
* [ ] **Task 316** - Create POST /api/audit-log/restore endpoint (restore previous version)
* [ ] **Task 317** - Create Audit Log Page (frontend)
* [ ] **Task 318** - Build comprehensive table:
  + Columns: Timestamp, User, Action, Linked Table, Linked Record ID, Old Value, New Value, Comments
* [ ] **Task 319** - Add filters:
  + Date range, User, Action type (Create, Edit, Delete, Link, Unlink, Split), Linked table
* [ ] **Task 320** - Implement "View Details" modal:
  + Show full old/new value JSON, Formatted display
* [ ] **Task 321** - Add "Restore" functionality:
  + Click "Restore" to revert to previous version
  + Exception: Cannot restore if transaction was split (data integrity)
  + Confirmation dialog
* [ ] **Task 322** - Add export functionality

**🎨 PHASE 3: UI/UX POLISH & NAVIGATION**

**3.1 Main Navigation & Layout**

* [ ] **Task 323** - Create main layout structure with sidebar navigation
* [ ] **Task 324** - Add WSC Sports logo above navigation
* [ ] **Task 325** - Build navigation menu structure:
  + Reports (Transactions, Trips, Travel Spend)
  + Data Integrity (Controls, Matching Engine, Split Engine)
  + Settings (Lists, Transformation, Countries/Cities, Tax, Owners)
  + Other (Audit Log)
* [ ] **Task 326** - Implement active state highlighting
* [ ] **Task 327** - Add responsive design for mobile/tablet
* [ ] **Task 328** - Create breadcrumb navigation
* [ ] **Task 329** - Add user profile menu (top-right)

**3.2 Common UI Components Enhancement**

* [ ] **Task 330** - Ensure all tables support saved views
* [ ] **Task 331** - Ensure all tables support column resize
* [ ] **Task 332** - Ensure all tables support column reorder
* [ ] **Task 333** - Ensure all tables support sort by any column
* [ ] **Task 334** - Add loading states to all data operations
* [ ] **Task 335** - Standardize all confirmation dialogs
* [ ] **Task 336** - Standardize all success/error notifications
* [ ] **Task 337** - Add keyboard shortcuts for common actions
* [ ] **Task 338** - Implement search highlighting

**3.3 Accessibility**

* [ ] **Task 339** - Add ARIA labels to all interactive elements
* [ ] **Task 340** - Ensure keyboard navigation works throughout
* [ ] **Task 341** - Test with screen readers
* [ ] **Task 342** - Add focus indicators
* [ ] **Task 343** - Ensure color contrast meets WCAG standards

**🧪 PHASE 4: TESTING**

**4.1 Unit Tests**

* [ ] **Task 344** - Test tax calculation logic
* [ ] **Task 345** - Test date formatting functions
* [ ] **Task 346** - Test number formatting functions
* [ ] **Task 347** - Test category mapping rules
* [ ] **Task 348** - Test split transaction logic
* [ ] **Task 349** - Test trip suggestion clustering algorithm
* [ ] **Task 350** - Test matching suggestions algorithm
* [ ] **Task 351** - Test amount calculations
* [ ] **Task 352** - Test validation utilities
* [ ] **Task 353** - Test message template generation

**4.2 Integration Tests**

* [ ] **Task 354** - Test CSV import with Navan sample data
* [ ] **Task 355** - Test CSV import with Agent sample data
* [ ] **Task 356** - Test link transaction to trip workflow
* [ ] **Task 357** - Test unlink transaction from trip workflow
* [ ] **Task 358** - Test split transaction creates correct records
* [ ] **Task 359** - Test manual trip creation workflow
* [ ] **Task 360** - Test trip suggestions generation
* [ ] **Task 361** - Test trip validation workflow
* [ ] **Task 362** - Test matching engine (manual and auto)
* [ ] **Task 363** - Test audit log captures all changes
* [ ] **Task 364** - Test restore functionality
* [ ] **Task 365** - Test filter combinations on all tables
* [ ] **Task 366** - Test sort functionality on all tables
* [ ] **Task 367** - Test export functions produce valid files

**4.3 UI/UX Tests**

* [ ] **Task 368** - Test all forms validate correctly
* [ ] **Task 369** - Test tables resize/reorder/sort
* [ ] **Task 370** - Test message templates generate correctly
* [ ] **Task 371** - Test confirmations work for all deletes
* [ ] **Task 372** - Test PDF declarations generate correctly
* [ ] **Task 373** - Test inline editing works properly
* [ ] **Task 374** - Test filters persist across page refreshes
* [ ] **Task 375** - Test modal dialogs display correctly
* [ ] **Task 376** - Test responsive design on different screen sizes

**4.4 Performance Tests**

* [ ] **Task 377** - Test large dataset handling (10,000+ transactions)
* [ ] **Task 378** - Test pagination performance
* [ ] **Task 379** - Test search/filter performance
* [ ] **Task 380** - Test export performance with large datasets
* [ ] **Task 381** - Optimize slow queries
* [ ] **Task 382** - Implement caching where appropriate

**4.5 Security Tests**

* [ ] **Task 383** - Test authentication and authorization
* [ ] **Task 384** - Test SQL injection prevention
* [ ] **Task 385** - Test XSS prevention
* [ ] **Task 386** - Test CSRF protection
* [ ] **Task 387** - Test file upload validation
* [ ] **Task 388** - Test API rate limiting

**🚀 PHASE 5: DEPLOYMENT & DOCUMENTATION**

**5.1 Database Initialization**

* [ ] **Task 389** - Create production database schema
* [ ] **Task 390** - Insert default lookup values
* [ ] **Task 391** - Create all required indexes
* [ ] **Task 392** - Set up database backup strategy
* [ ] **Task 393** - Set up audit triggers (optional)

**5.2 First Run Setup**

* [ ] **Task 394** - Import Countries & Cities CSV
* [ ] **Task 395** - Import Tax rates CSV
* [ ] **Task 396** - Import Headcount data
* [ ] **Task 397** - Set up Owners
* [ ] **Task 398** - Configure transformation rules
* [ ] **Task 399** - Import historical transactions/trips

**5.3 Documentation**

* [ ] **Task 400** - Write API documentation
* [ ] **Task 401** - Write user manual
* [ ] **Task 402** - Create admin guide
* [ ] **Task 403** - Document CSV import formats
* [ ] **Task 404** - Document transformation rules configuration
* [ ] **Task 405** - Create troubleshooting guide
* [ ] **Task 406** - Write deployment guide
* [ ] **Task 407** - Document backup and recovery procedures

**5.4 Deployment**

* [ ] **Task 408** - Set up production environment
* [ ] **Task 409** - Configure production database
* [ ] **Task 410** - Set up Snowflake connection
* [ ] **Task 411** - Configure environment variables
* [ ] **Task 412** - Deploy application
* [ ] **Task 413** - Set up monitoring and logging
* [ ] **Task 414** - Configure automated backups
* [ ] **Task 415** - Set up error tracking (Sentry, etc.)
* [ ] **Task 416** - Configure SSL certificates
* [ ] **Task 417** - Set up CDN (if applicable)

**5.5 Training & Handoff**

* [ ] **Task 418** - Conduct user training sessions
* [ ] **Task 419** - Create video tutorials
* [ ] **Task 420** - Set up help desk/support system
* [ ] **Task 421** - Document known issues and workarounds
* [ ] **Task 422** - Create FAQ document
* [ ] **Task 423** - Hand off to support team

**📈 PHASE 6: FUTURE ENHANCEMENTS (Post-MVP)**

**6.1 Advanced Features**

* [ ] **Task 424** - Implement OCR for receipt scanning
* [ ] **Task 425** - Add email integration for automatic message sending
* [ ] **Task 426** - Create mobile app for expense submission
* [ ] **Task 427** - Integrate real-time exchange rate API
* [ ] **Task 428** - Build dashboard with charts and KPIs
* [ ] **Task 429** - Implement approval workflow system
* [ ] **Task 430** - Add budget tracking feature
* [ ] **Task 431** - Implement notifications and reminders
* [ ] **Task 432** - Add multi-currency support improvements
* [ ] **Task 433** - Create advanced analytics and reporting

**6.2 Integrations**

* [ ] **Task 434** - Integrate with accounting system (QuickBooks, etc.)
* [ ] **Task 435** - Integrate with HR system for headcount sync
* [ ] **Task 436** - Integrate with Slack for notifications
* [ ] **Task 437** - Integrate with Google Drive for document storage
* [ ] **Task 438** - Add SSO integration

**📝 TASK REFERENCE GUIDE**

**How to use this list with Copilot:**

1. **Reference by number**: "Work on Task 156" or "Complete tasks 074-081"
2. **Group tasks**: "Implement all backend APIs for Transactions (Tasks 074-081)"
3. **Track progress**: Mark tasks as complete with ✅
4. **Prioritize**: Mark urgent tasks with ⚡
5. **Dependencies**: Note "Requires Task 031" for dependent tasks

**Estimation Guide:**

* Simple tasks (data models, utility functions): 1-2 hours
* Medium tasks (API endpoints, simple UI pages): 2-4 hours
* Complex tasks (algorithms, complex UIs): 4-8 hours
* Very complex tasks (full features): 1-2 days

**Total Tasks: 438**

* Phase 0 (Foundation): Tasks 001-020 (20 tasks)
* Phase 1 (Database): Tasks 021-048 (28 tasks)
* Phase 2 (Services): Tasks 049-073 (25 tasks)
* Tab 1 (Reports): Tasks 074-156 (83 tasks)
* Tab 2 (Data Integrity): Tasks 157-257 (101 tasks)
* Tab 3 (Settings): Tasks 258-322 (65 tasks)
* Phase 3 (UI/UX): Tasks 323-343 (21 tasks)
* Phase 4 (Testing): Tasks 344-388 (45 tasks)
* Phase 5 (Deployment): Tasks 389-423 (35 tasks)
* Phase 6 (Future): Tasks 424-438 (15 tasks)

**END OF TODO LIST**

*For questions or clarifications on any task, refer back to the FinalProposal.docx specification document.*

* [ ] **Task 293** - Create GET /api/settings/tax endpoint
* [ ] **Task 294** - Create POST /api/settings/tax/import endpoint (CSV upload)
* [ ] **Task 295** - Create POST /api/settings/tax endpoint (add manual entry)
* [ ] **Task 296** - Create PUT /api/settings/tax/:id endpoint (update)
* [ ] **Task 297** - Create DELETE /api/settings/tax/:id endpoint (delete)
* [ ] **Task 298** - Create Tax Settings Page (frontend)
* [ ] **Task 299** - Build import section:
  + File upload for Tax CSV
  + Import button
  + Progress indicator
* [ ] **Task 300** - Create table:
  + Columns: Fiscal Year, Country, Subsidiary, Meals Cap, Lodging Cap, Tax Shield, Actions
* [ ] **Task 301** - Add filters: Fiscal Year, Country, Subsidiary
* [ ] **Task 302** - Add "Add Manual Entry" button
* [ ] **Task 303** - Create Add/Edit modal:
  + Fiscal Year input (4-digit)
  + Country dropdown
  + Subsidiary input
  + Meals Cap, Lodging Cap, Tax Shield inputs (numeric with validation)
  + Save/Cancel buttons
* [ ] **Task 304** - Add validation: All cap amounts must be positive numbers
* [ ] **Task 305** - Add export functionality

**3.5 Settings - Owners Management**

* [ ] **Task 306** - Create GET /api/settings/owners endpoint
* [ ] **Task 307** - Create POST /api/settings/owners endpoint (add owner)
* [ ] **Task 308** - Create PUT /api/settings/owners/:id endpoint (update)
* [ ] **Task 309** - Create DELETE /api/settings/owners/:id endpoint (delete)
* [ ] **Task 310** - Create Owners Management Page (frontend)
* [ ] **Task 311** - Build table:
  + Columns: Cost Center, Department, Domain, Owner(s), Actions
* [ ] **Task 312** - Add "Add New Owner" button
* [ ] **Task 313** - Create Add/Edit modal:
  + Cost Center input
  + Department input
  + Domain input
  + Owner multi-select/dropdown
  + Save/Cancel buttons
* [ ] **Task 314** - Add delete confirmation

**3.6 Audit Log**

* [ ] **Task 315** - Create GET /api/audit-log endpoint (with pagination and filters)
* [ ] **Task 316** - Create POST /api/audit-log/restore endpoint (restore previous version)
* [ ] **Task 317** - Create Audit Log Page (frontend)
* [ ] **Task 318** - Build comprehensive table:
  + Columns: Timestamp, User, Action, Linked Table, Linked Record ID, Old Value, New Value, Comments
* [ ] **Task 319** - Add filters:
  + Date range
  + User
  + Action type (Create, Edit, Delete, Link, Unlink, Split)
  + Linked table
* [ ] **Task 320** - Implement "View Details" modal:
  + Show full old/new value JSON
  + Formatted display
* [ ] **Task 321** - Add "Restore" functionality:
  + Click "Restore" to revert to previous version
  + Exception: Cannot restore if transaction was split (data integrity)
  + Confirmation dialog
* [ ] **Task 322** - Add export functionality
* [ ] **Backend API**