

PassivePilot Dashboard Redesign Documentation

Overview

This document describes the complete dashboard redesign for PassivePilot v3, featuring a modern black & gold theme with enhanced user experience and Mapbox GL JS integration.

Design Specifications

Color Palette

- **Background:** #0a0a0a (near black)
- **Cards:** #1a1a1a (dark gray)
- **Borders:** #262626 (medium gray)
- **Primary Gold:** #d4af37 (classic gold)
- **Secondary Gold:** #f59e0b (bright amber)
- **Text:** #ffffff (white) and #9ca3af (gray-400)

Typography

- **Font Families:** Inter (sans-serif), Playfair Display (serif), Montserrat, Poppins
- **Headings:** Bold, white color
- **Body Text:** Regular weight, gray-400 for secondary text

Pages Implemented

1. Dashboard (/dashboard)

Route: /app/dashboard/page.tsx

Purpose: Main command center showing property overview, stats, and map visualization

Components Used:

- DashboardStats - Top stats bar showing Deal Count Verified, Assignment Received, Slip per Fixer
- PropertyGrid - Grid of property cards with images, details, and "Show More" buttons
- MapPanel - Mapbox map showing property locations
- Sidebar - Right sidebar with tool links

API Integration:

- GET /api/deals - Fetches all deals
- GET /api/dashboard/stats - Fetches dashboard statistics

Features:

- Responsive grid layout (7 columns for properties, 5 columns for map/sidebar)
- Real-time data loading with loading states
- Property cards display: owner name, address, price, discount %, beds, baths, sqft
- Map markers for each property with latitude/longitude
- Tool links in sidebar for quick navigation

2. Property Search (/properties/search)

Route: /app/properties/search/page.tsx

Purpose: Search and filter properties with interactive map

Components Used:

- `PropertyListItem` - Individual property in search results list
- `MapPanel` - Large map view with property markers
- Filter bar with Search>Select>Analyze>Contact/Connect steps

API Integration:

- GET /api/properties/search - Search properties with filters

Features:

- Multi-step workflow indicator (Search → Select → Analyze → Contact → Connect)
- Left panel: Scrollable property list (4 columns)
- Right panel: Large interactive map (8 columns)
- Click property in list → highlights marker on map
- Click marker on map → selects property in list
- Search input with filter button
- Custom gold/orange markers (default) and black with gold border (selected)

3. Deal Analysis (/deals/analyze/[id])

Route: /app/deals/analyze/[id]/page.tsx

Purpose: Detailed analysis of a single deal with ARV, deal score, and property metadata

Components Used:

- `DealAnalysisCard` - Left panel with property photo, ARV, assignment fee, dispose agent, deal score
- `PropertyMetadata` - Middle panel with owner info, property details, tabs, estimations
- `MapPanel` - Right panel with single property marker

API Integration:

- GET /api/deals/{id} - Fetch single deal details

Features:

- Back to Dashboard button
- Circular property photo with gold border
- ARV display in large gold text
- Assignment Fee with toggle switch
- Dispose Agent with toggle switch
- Deal Score progress bar (0-100%) with gold gradient
- Property details: beds, baths, sqft, year built, lot size, county
- Tabs: Essentials (active), Comestication, O-Comestible
- Estimations breakdown (repair cost, holding cost, closing cost)
- Labels: "Essentials" (gold), "Estimated" (green)
- Map with single property pin

Component Hierarchy

/components/dashboard/	
└── ARVDisplay.tsx	- ARV value display with large gold text
└── DashboardStats.tsx	- Top statistics cards
└── DealAnalysisCard.tsx	- Deal analysis left panel
└── DealScoreBar.tsx	- Progress bar for deal score (0-100%)
└── MapPanel.tsx	- Mapbox GL JS map component
└── PropertyCard.tsx	- Property card with image and details
└── PropertyGrid.tsx	- Grid layout for property cards
└── PropertyListItem.tsx	- Property item in search results
└── PropertyMetadata.tsx	- Property details and metadata display
└── Sidebar.tsx	- Right sidebar with tool links

Component Details

MapPanel

- Uses Mapbox GL JS (not Leaflet)
- Requires `NEXT_PUBLIC_MAPBOX_TOKEN` environment variable
- Custom markers: gold/orange circles (default), black with gold border (selected)
- Popups show property address, price, ARV, discount, “View Details” button
- Hover effect: markers scale up on hover
- Fit bounds to show all markers automatically
- Click marker → trigger `onMarkerClick` callback

DashboardStats

- Grid layout: 3 columns on desktop, 1 column on mobile
- Each stat card: dark background, border, large text for value

PropertyCard

- Horizontal layout: image on left, details on right
- Shows: owner name, address, beds, baths, sqft, price, discount %
- “Show More” button links to `/deals/analyze/[id]`
- Hover effect: border changes to gold

DealAnalysisCard

- Vertical layout: photo at top, then ARV, fee, agent, score
- Circular property photo with gold border
- Toggle switches for assignment fee and dispose agent
- Deal score progress bar with gradient (gold to amber)
- Labels at bottom: “Essentials”, “Estimated”

API Integration

Base Configuration

```
// Environment Variables
NEXT_PUBLIC_API_BASE_URL=http://127.0.0.1:8000
NEXT_PUBLIC_MAPBOX_TOKEN=your_mapbox_token_here
```

API Functions

Located in `/lib/api.ts` :

```
// Fetch all deals
fetchDeals(): Promise<Deal[]>

// Fetch single deal
fetchDeal(id: string): Promise<Deal>

// Analyze deal
analyzeDeal(payload: { dealId: string }): Promise<DealAnalysis>

// Fetch campaigns
fetchCampaigns(): Promise<Campaign[]>

// Fetch dashboard stats
fetchDashboardStats(): Promise<DashboardStats>

// Search properties
searchProperties(filters?: {...}): Promise<Property[]>
```

Error Handling

- All API functions include try/catch blocks
- Console logging for debugging
- Graceful fallbacks (empty arrays for lists, default values for stats)
- User-friendly error messages on UI

Environment Variables

Required

```
NEXT_PUBLIC_MAPBOX_TOKEN=pk.eyJ1IjoieW91c11c2VybmFtZSIsImEiOiJjbHh4eHh4eHgifQxxxxxxxxxxxxxxxxxxxx
```

Optional

```
NEXT_PUBLIC_API_BASE_URL=http://127.0.0.1:8000
NEXT_PUBLIC_API_BASE=http://127.0.0.1:8000
```

If not set, defaults to `http://127.0.0.1:8000`

Theme Customization

Updating Colors

Edit `/code/frontend/app/globals.css` :

```
.dark {
  --background: 0 0% 4%;      /* #0a0a0a */
  --card: 0 0% 10%;          /* #1a1a1a */
  --border: 0 0% 15%;        /* #262626 */
  --primary: 43 74% 52%;    /* #d4af37 */
  --secondary: 38 100% 51%;  /* #f59e0b */
}
```

Edit /code/frontend/tailwind.config.ts :

```
colors: {
  gold: {
    500: '#D4AF37', // Primary gold
    600: '#C9A55E',
    // ... other shades
  },
}
```

Custom Animations

Available animations in globals.css:

- `animate-fade-in-up` - Fade in with upward motion
- `animate-scale-pop` - Scale and pop effect
- `btn-glow-pulse` - Glowing pulse for buttons
- `btn-shimmer` - Shimmer effect on hover

Responsive Design

Breakpoints (Tailwind defaults)

- `sm` : 640px
- `md` : 768px
- `lg` : 1024px
- `xl` : 1280px
- `2xl` : 1536px

Layout Adjustments

Dashboard:

- Desktop: 7 columns (properties) + 5 columns (map/sidebar)
- Tablet/Mobile: Stack vertically, full width

Property Search:

- Desktop: 4 columns (list) + 8 columns (map)
- Mobile: List first, then map (full width, toggle view)

Deal Analysis:

- Desktop: 4 columns (analysis) + 5 columns (details) + 3 columns (map)
- Mobile: Stack vertically

Development Workflow

Setup

```
cd code/frontend
npm install
```

Run Development Server

```
npm run dev
```

Open <http://localhost:3000> (`http://localhost:3000`)

Build for Production

```
npm run build
npm run start
```

Linting

```
npm run lint
```

Future Enhancements

Backend Requirements

The following API endpoints may need to be implemented or enhanced:

1. **Deal Scoring:** `POST /api/deals/score` - Calculate deal score based on property metrics
2. **Property Images:** Integration with ATTOM or external image sources
3. **Geolocation:** Geocoding service for properties without lat/lng
4. **Advanced Filters:** `/api/properties/search` with complex filter support
5. **Campaign Management:** Full CRUD operations for campaigns

Frontend Enhancements

1. **Property Filters:** Advanced filter panel with price range, beds, baths, sqft, year built
2. **Map Clustering:** Cluster markers when zoomed out (Mapbox supercluster)
3. **Property Comparison:** Side-by-side comparison of multiple properties
4. **Export Reports:** Generate PDF reports of deal analysis
5. **Saved Searches:** Save and reuse search criteria
6. **Notifications:** Real-time alerts for new deals matching criteria
7. **Mobile App:** React Native version with same design system

Troubleshooting

Map Not Displaying

Issue: Map shows “Map Unavailable” message

Solution: Set `NEXT_PUBLIC_MAPBOX_TOKEN` in `.env.local`

```
NEXT_PUBLIC_MAPBOX_TOKEN=your_token_here
```

API Errors

Issue: “Failed to fetch deals” or similar errors

Solutions:

1. Check backend is running: `http://127.0.0.1:8000/docs`
2. Verify `NEXT_PUBLIC_API_BASE_URL` is correct
3. Check browser console for detailed error messages
4. Ensure CORS is configured on backend

Build Errors

Issue: TypeScript errors during build

Solution: Run lint first to identify issues

```
npm run lint
npm run build
```

Image Loading Errors

Issue: Next.js Image component fails to load

Solutions:

1. Add image domains to `next.config.js`:

```
images: {
  domains: ['your-cdn-domain.com'],
}
```

1. Use `unoptimized: true` for external images (not recommended for production)

Testing

Manual Testing Checklist

- [] Dashboard loads with stats and properties
- [] Map displays markers for all properties
- [] Click property card → navigate to deal analysis
- [] Property search filters work
- [] Click property in list → highlights on map
- [] Click marker on map → selects property
- [] Deal analysis shows all details correctly
- [] Back button returns to dashboard
- [] Responsive layout works on mobile
- [] Dark theme colors match template

Support

For issues or questions:

- GitHub Issues: [Repository Issues](https://github.com/MohamedA12289/Passive_Pilot/issues) (https://github.com/MohamedA12289/Passive_Pilot/issues)
- Documentation: See this file and code comments
- Backend API Docs: <http://127.0.0.1:8000/docs> (when running locally)