Complete Trustworthy Implementation Guide

Based on Video Analysis - Exact Technical Structure

🎯 Complete Workflow from Video

User Journey:

1. **Click Browse** → File picker opens
2. **Select file** → Upload starts with progress
3. **LEFT sidebar opens** → Document appears with thumbnail
4. **AI analyzes** → "Analyzing..." spinner shows
5. **Lightning bolt appears** → "⚡ Details 2" button on thumbnail
6. **Click Details** → Modal opens with filename suggestion
7. **Accept/Dismiss** → Filename suggestion handling
8. **Route to profile** → Angel Quintana's family member page
9. **Document sections** → Driver's License, Passport upload areas

🏗️ Frontend Structure

1. Main Components

JSX

```jsx
// Main Upload Center Component
components/
├── UploadCenter/
│   ├── TrustworthyUploadCenter.jsx     // Main container
│   ├── UploadZone.jsx                  // Browse button area
│   ├── LeftSidebar.jsx                 // Document inbox (slides from LEFT)
│   ├── DocumentItem.jsx               // Individual document with lightning
bolt
│   ├── FilenameModal.jsx               // Suggestion modal
│   ├── FamilyMemberProfile.jsx         // Profile destination page
│   └── DocumentSections.jsx           // Driver's License, Passport areas
```

2. State Management

JSX

```jsx
// Main state structure
const [uploadState, setUploadState] = useState({
  currentStep: 'browse', // browse, uploading, analyzing, details_ready, modal_open, profile_view
  uploadProgress: 0,
  uploadedDocument: null,
  extractedData: null,
  isLeftSidebarOpen: false,
  isFilenameModalOpen: false,
  selectedFamilyMember: null,
  suggestedFilename: ''
});

// Document structure
const documentStructure = {
  id: 'doc_123',
  filename: 'screenshot_20250906_164837_gmail.jpg',
  thumbnail: 'blob:url',
  uploadTime: new Date(),
  analysisResult: {
    extractedFields: [
      { key: "Person Name", value: "ANGEL D QUINTANA", confidence: 95 },
      { key: "Social Security Number", value: "141-85-2645", confidence: 92 }
    ],
    documentType: "Social Security Card",
    identifiedPerson: {
      name: "Angel Quintana",
      id: "person_angel",
      category: "Family IDs",
      subcategory: "Family Member"
    },
    suggestedFilename: "Social Security Card Angel Quintana"
  }
};
```

3. Key Frontend Functions

JSX

```jsx
// File upload handler
const handleFileUpload = async (files) => {
  const file = files[0];

  // Step 1: Start upload
  setUploadState(prev => ({ ...prev, currentStep: 'uploading' }));

  // Step 2: Upload file
  const uploadResult = await uploadFile(file);

  // Step 3: Open LEFT sidebar
  setUploadState(prev => ({
    ...prev,
    currentStep: 'analyzing',
    uploadedDocument: uploadResult,
    isLeftSidebarOpen: true
  }));

  // Step 4: Start AI analysis
  const analysisResult = await analyzeDocument(uploadResult.id);

  // Step 5: Show lightning bolt
  setUploadState(prev => ({
    ...prev,
    currentStep: 'details_ready',
    extractedData: analysisResult
  }));
};

// Lightning bolt click handler
const handleDetailsClick = () => {
  setUploadState(prev => ({
    ...prev,
    currentStep: 'modal_open',
    isFilenameModalOpen: true
  }));
};

// Filename acceptance handler
const handleFilenameAccept = () => {
  setUploadState(prev => ({
    ...prev,
    currentStep: 'profile_view',
    isFilenameModalOpen: false,
    selectedFamilyMember: extractedData.identifiedPerson
  }));
};
```

🔧 Backend Structure

1. API Endpoints

JavaScript

```
// Upload endpoint
POST /api/upload
{
  file: FormData,
  metadata: {
    familyId: 'family_123',
    uploadedBy: 'user_456'
  }
}

Response: {
  success: true,
  data: {
    fileId: 'doc_789',
    filename: 'original_name.jpg',
    thumbnail: 'https://cdn.../thumbnail.jpg',
    uploadUrl: 'https://cdn.../document.jpg'
  }
}

// AI Analysis endpoint
POST /api/analyze-document
{
  fileId: 'doc_789',
  analysisType: 'full_extraction'
}

Response: {
  success: true,
  data: {
    extractedFields: [
      { key: "Person Name", value: "ANGEL D QUINTANA", confidence: 95 },
      { key: "Social Security Number", value: "141-85-2645", confidence: 92 }
    ],
    documentType: "Social Security Card",
    identifiedPerson: {
      name: "Angel Quintana",
      id: "person_angel",
      matchConfidence: 88
    },
    suggestedFilename: "Social Security Card Angel Quintana",
    documentCategory: "identity_document"
  }
}
```

```
// Family member endpoints
GET /api/family-members/:id
POST /api/family-members/:id/documents
PUT /api/documents/:id/filename
```

2. Database Schema

SQL

```sql
-- Documents table
CREATE TABLE documents (
  id VARCHAR(255) PRIMARY KEY,
  family_id VARCHAR(255) NOT NULL,
  filename VARCHAR(500) NOT NULL,
  original_filename VARCHAR(500) NOT NULL,
  file_url TEXT NOT NULL,
  thumbnail_url TEXT,
  file_size BIGINT,
  mime_type VARCHAR(100),
  upload_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
  analysis_status ENUM('pending', 'processing', 'completed', 'failed'),
  analysis_result JSON,
  assigned_to_person VARCHAR(255),
  document_category VARCHAR(100),
  INDEX idx_family_id (family_id),
  INDEX idx_assigned_person (assigned_to_person)
);

-- Family members table
CREATE TABLE family_members (
  id VARCHAR(255) PRIMARY KEY,
  family_id VARCHAR(255) NOT NULL,
  name VARCHAR(255) NOT NULL,
  birthday DATE,
  profile_data JSON,
  created_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
  INDEX idx_family_id (family_id)
);

-- Document analysis table
CREATE TABLE document_analysis (
  id VARCHAR(255) PRIMARY KEY,
  document_id VARCHAR(255) NOT NULL,
  extracted_fields JSON,
  confidence_scores JSON,
  person_matches JSON,
  suggested_filename VARCHAR(500),
  document_type VARCHAR(100),
  analysis_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
  FOREIGN KEY (document_id) REFERENCES documents(id)
);
```

3. AI Analysis Service

JavaScript

```javascript
// AI Analysis Service
class DocumentAnalysisService {
  async analyzeDocument(fileId, fileUrl) {
    try {
      // Step 1: Extract text using AWS Textract or similar
      const textractResult = await this.extractText(fileUrl);

      // Step 2: Identify document type
      const documentType = await this.classifyDocument(textractResult);

      // Step 3: Extract structured fields
      const extractedFields = await this.extractFields(textractResult,
documentType);

      // Step 4: Identify person from extracted data
      const personMatch = await this.identifyPerson(extractedFields);

      // Step 5: Generate smart filename
      const suggestedFilename = this.generateFilename(documentType,
personMatch, extractedFields);

      return {
        extractedFields,
        documentType,
        identifiedPerson: personMatch,
        suggestedFilename,
        confidence: this.calculateOverallConfidence(extractedFields)
      };
    } catch (error) {
      throw new Error(`Analysis failed: ${error.message}`);
    }
  }

  async extractText(fileUrl) {
    // AWS Textract integration
    const textract = new AWS.Textract();
    const params = {
      Document: { S3Object: { Bucket: 'your-bucket', Name: fileUrl } },
      FeatureTypes: ['FORMS', 'TABLES']
    };
    return await textract.analyzeDocument(params).promise();
  }

  classifyDocument(textractResult) {
    // Document classification logic
    const text = this.extractTextFromResult(textractResult);

    if (text.includes('SOCIAL SECURITY')) return 'Social Security Card';
    if (text.includes('DRIVER LICENSE')) return 'Driver License';
    if (text.includes('PASSPORT')) return 'Passport';

    return 'Unknown Document';
  }

  extractFields(textractResult, documentType) {
    // Field extraction based on document type
    const fields = [];

    switch (documentType) {
      case 'Social Security Card':
        fields.push(
          this.extractField(textractResult, 'name', /[A-Z\s]{2,}/),
          this.extractField(textractResult, 'ssn', /\d{3}-\d{2}-\d{4}/)
        );
        break;
      // Add other document types
    }

    return fields.filter(field => field.value);
  }

  async identifyPerson(extractedFields) {
    // Person identification logic
    const nameField = extractedFields.find(f =>
f.key.toLowerCase().includes('name'));
    if (!nameField) return null;

    // Search existing family members
    const existingMember = await this.findFamilyMember(nameField.value);
    if (existingMember) {
      return {
        ...existingMember,
        matchConfidence: this.calculateNameMatch(nameField.value,
existingMember.name)
      };
    }
    // Suggest new family member
    return {
      name: this.formatName(nameField.value),
      id: null,
      isNew: true,
      category: 'Family IDs',
      subcategory: 'Family Member'
    };
  }

  generateFilename(documentType, personMatch, extractedFields) {
    if (personMatch && documentType) {
      return `${documentType} ${personMatch.name}`;
    }
    return `Document ${new Date().toISOString().split('T')[0]}`;
  }
}
```

🎨 CSS Structure (Dark Gold Theme)

CSS

```css
/* Main upload center styles */
.trustworthy-upload-center {
  display: flex;
  min-height: 100vh;
  background: #0F0F0F;
  color: #E5E7EB;
  font-family: Inter, sans-serif;
}

/* LEFT sidebar - key correction from video */
.trustworthy-left-sidebar {
  position: fixed;
  top: 0;
  left: 0;  /* LEFT side, not right */
  width: 400px;
  height: 100vh;
  background: #1a1a1a;
  border-right: 1px solid #333333;
  transform: translateX(-100%);
  transition: transform 0.3s ease;
  z-index: 1000;
}
```

```css
.trustworthy-left-sidebar.open {
  transform: translateX(0);
}

/* Lightning bolt button on thumbnail */
.lightning-bolt-button {
  position: absolute;
  bottom: 8px;
  left: 8px;
  background: #F59E0B;
  color: #000;
  border: none;
  padding: 4px 8px;
  border-radius: 6px;
  font-size: 12px;
  font-weight: 600;
  display: flex;
  align-items: center;
  gap: 4px;
  cursor: pointer;
  box-shadow: 0 2px 4px rgba(0,0,0,0.3);
}

/* Filename modal */
.filename-modal {
  position: fixed;
  top: 50%;
  left: 50%;
  transform: translate(-50%, -50%);
  background: #1a1a1a;
  border: 1px solid #333333;
  border-radius: 12px;
  padding: 24px;
  max-width: 500px;
  width: 90%;
  z-index: 2000;
}

/* Family member profile */
.family-member-profile {
  background: #0F0F0F;
  min-height: 100vh;
  padding: 24px;
}

.document-upload-area {
  border: 2px dashed #333333;
  border-radius: 8px;
  padding: 24px;
  text-align: center;
  background: rgba(26, 26, 26, 0.5);
  transition: all 0.3s ease;
}

.document-upload-area:hover {
  border-color: #D4AF37;
  background: rgba(212, 175, 55, 0.05);
}
```

🔄 Complete Implementation Flow

1. File Upload Process

JavaScript

```
// Frontend trigger
handleBrowseClick() → fileInput.click() → handleFileUpload()

// Backend process
POST /api/upload → S3 upload → thumbnail generation → database record

// Frontend update
setUploadState({ currentStep: 'analyzing', isLeftSidebarOpen: true })
```

2. AI Analysis Process

JavaScript

```
// Backend trigger
POST /api/analyze-document → AWS Textract → field extraction → person
matching

// Frontend update
setUploadState({ currentStep: 'details_ready', extractedData: result })
```

3. User Interaction Process

JavaScript

```
// Lightning bolt click
handleDetailsClick() → setUploadState({ isFilenameModalOpen: true })
```

```
// Filename acceptance
handleFilenameAccept() → PUT /api/documents/:id/filename → profile routing
```

📦 Required Dependencies

JSON

```json
{
  "frontend": {
    "react": "^18.0.0",
    "framer-motion": "^10.0.0",
    "react-dropzone": "^14.0.0"
  },
  "backend": {
    "express": "^4.18.0",
    "multer": "^1.4.5",
    "aws-sdk": "^2.1400.0",
    "mysql2": "^3.6.0",
    "sharp": "^0.32.0"
  }
}
```

🚀 Integration Steps

1. **Set up file upload** with progress tracking
2. **Implement LEFT sidebar** with document display
3. **Add AI analysis** with AWS Textract integration
4. **Create lightning bolt** button with field count
5. **Build filename modal** with accept/dismiss
6. **Design family member profiles** with document sections
7. **Connect routing** from upload to profiles

This gives you the complete technical structure to implement the exact Trustworthy workflow on your app page with your dark gold theme.