Nigerian Taxbot Documentation

Table of Contents

Core Functionalities

1. Tax Information and Guidance

- Provides accurate information on Nigerian tax regulations and policies
- Explains tax concepts in conversational, easy-to-understand language
- Covers multiple tax types including VAT, CIT, PIT, CGT, and WHT
- References up-to-date tax rates, deadlines, and

compliance requirements

2. VAT Calculation

- Calculates 7.5% Value Added Tax on goods and services
- Processes natural language input to extract prices and quantities
- Provides detailed breakdown of base amount, VAT amount, and total
- Handles unit prices and quantities for itemized calculations

3. VAT Exemption Checking

- Identifies products and services exempt from VAT
- References comprehensive database of VAT-exempt categories
- Provides regulatory basis for exemptions
- Handles synonym matching for more accurate exemption identification

4. Tax Calculator Capabilities

- Companies Income Tax (CIT) calculations based on company size and turnover
- Personal Income Tax (PIT) calculations with

progressive tax bands

- Capital Gains Tax (CGT) calculations for property and asset sales
- Pay As You Earn (PAYE) tax calculations for employees

User Experience Features

1. Interactive Chat Interface

- Real-time message exchange with animated typing indicators
- Message history preservation during session
- Copy functionality for message content
- Loading states and error handling for API communication

2. Error Management

- Fallback responses when API connection fails
- Clear error messages for troubleshooting
- Retry mechanisms for failed requests
- Graceful degradation with local processing when possible

3. Example Queries

- Suggested tax-related questions for new users
- Quick-access buttons for common tax inquiries
- Educational prompts to guide user interaction

4. Responsive Design

- Mobile-friendly interface
- Adaptable layout for different screen sizes
- Accessible design elements for improved usability

Technical Infrastructure

1. API Integration

```
// Example of API integration
interface APIConfig {
  endpoint: string;
  apiKey: string;
  model: string;
  maxTokens: number;
}

const mistralConfig: APIConfig = {
  endpoint:
"https://api.mistral.ai/v1/chat/completions",
  apiKey: process.env.MISTRAL_API_KEY,
  model: "mistral-tiny",
  maxTokens: 1024
};
```

2. Data Processing

```
// Example of data processing
interface TaxCalculation {
  baseAmount: number;
  taxRate: number;
  taxAmount: number;
  totalAmount: number;
}

const calculateTax = (amount: number, rate:
number): TaxCalculation => {
  const taxAmount = amount * rate;
  return {
    baseAmount: amount,
    taxRate: rate,
    taxAmount: taxAmount,
    totalAmount: amount + taxAmount
  };
};
```

Information Flow

1. User Input Processing

```
interface UserMessage {
  role: "user";
  content: string;
  metadata?: {
    timestamp: number;
    sessionId: string;
  };
}

interface AssistantMessage {
  role: "assistant";
  content: string;
  metadata?: {
    calculationType?: string;
    taxType?: string;
  };
}
```

2. Response Formatting

```
interface FormattedResponse {
  type: "calculation" | "information" | "error";
  content: string;
  metadata?: {
    taxType?: string;
    rate?: number;
    reference?: string;
  };
}
```

Response Processing

1. Message Pipeline

1. Input Processing
    - Message validation
    - Context analysis
    - Intent detection
    - Data extraction
2. Business Logic
    - Tax type determination
    - Rate application
    - Calculation execution
    - Exemption checking

3. Response Generation
    - Format selection
    - Content structuring
    - Reference inclusion
    - Presentation styling

2. Response Types

```
interface TaxResponse {
  type: "VAT" | "CIT" | "PIT" | "CGT";
  calculation?: {
    input: number;
    rate: number;
    result: number;
  };
  explanation: string;
  references?: string[];
}
```

Error Handling

1. API Error Handling

```
const handleAPIError = async (error: Error) => {
  console.error("API Error:", error);

  // Return fallback response
  return {
    type: "error",
    content: getFallbackResponse(),
    metadata: {
      error: error.message,
      timestamp: Date.now()
    }
  };
};
```

2. Fallback Responses

```
const getFallbackResponse = (query: string):
string => {
  const taxType = determineTaxType(query);
  return FALLBACK_RESPONSES[taxType] ||
DEFAULT_RESPONSE;
};
```

Future Enhancements

1. Planned Features

- Real-time tax rate updates
- Integration with tax filing systems
- Enhanced multilingual support
- Advanced calculation features

2. Technical Improvements

- Enhanced caching system
- Performance optimizations
- Additional API integrations
- Expanded error handling

Usage Examples

1. VAT Calculation

```
// User Query: "Calculate VAT for ₦50,000"
const response = {
  type: "calculation",
  content: `
VAT Calculation Results:
Base Amount: ₦50,000.00
VAT (7.5%): ₦3,750.00
Total Amount: ₦53,750.00
  `,
  metadata: {
    taxType: "VAT",
    rate: 0.075,
    reference: "FIRS VAT Act"
  }
};
```

2. Tax Information

```
// User Query: "What is the CIT rate for small companies?"
const response = {
  type: "information",
  content: `
Companies with annual turnover less than ₦25 million are considered small companies and are exempt from Companies Income Tax (CIT).

Reference: Companies Income Tax Act, Section 40(6)
  `,
  metadata: {
    taxType: "CIT",
    category: "small_business",
    lastUpdated: "2024"
  }
};
```

This documentation is maintained by the Taxbot development team. Last updated: March 2024