

Frontend Agent - Google ADK Implementation

A specialized AI agent built with Google's Agent Development Kit (ADK) that generates modern frontend web applications based on user descriptions.

Overview

The Frontend Agent is designed to create clean, responsive, and aesthetically pleasing HTML, CSS (using Tailwind CSS), and JavaScript code. It follows modern web development best practices and can generate complete web applications from simple text descriptions.

Features

- **Semantic HTML5 Generation:** Creates well-structured, accessible markup
- **Tailwind CSS Styling:** Uses Tailwind CSS for modern, responsive designs
- **Modern JavaScript:** Generates clean ES6+ JavaScript for interactivity
- **Component-Based Thinking:** Designs reusable, maintainable code
- **Code Validation:** Built-in validation and optimization suggestions
- **Multiple Templates:** Pre-built templates for common use cases

Installation

Prerequisites

- Python 3.9+ or Java 17+
- Google Cloud account or Google AI Studio account
- Terminal/Command line access

Step 1: Install Google ADK

```
# Create virtual environment (recommended)
python -m venv .venv

# Activate virtual environment
# macOS/Linux:
source .venv/bin/activate
# Windows CMD:
.venv\Scripts\activate.bat
# Windows PowerShell:
.venv\Scripts\Activate.ps1

# Install ADK
pip install google-adk
```

Step 2: Set up Authentication

Option A: Google AI Studio (Recommended for Development)

1. Go to [Google AI Studio](#)
2. Create an API key
3. Update the `.env` file:

```
GOOGLE_GENAI_USE_VERTEXAI=FALSE  
GOOGLE_API_KEY=your_actual_api_key_here
```

Option B: Google Cloud Vertex AI (For Production)

1. Set up a Google Cloud project
2. Enable Vertex AI API
3. Authenticate with gcloud:

```
gcloud auth application-default login
```

4. Update the `.env` file:

```
GOOGLE_GENAI_USE_VERTEXAI=TRUE  
GOOGLE_CLOUD_PROJECT=your_project_id  
GOOGLE_CLOUD_LOCATION=us-central1
```

Usage

Running the Agent

Navigate to the parent directory of the `frontend_agent` folder and run:

Web UI (Recommended)

```
adk web
```

Then open `http://localhost:8000` in your browser.

Command Line Interface

```
adk run frontend_agent
```

API Server

```
adk api_server frontend_agent
```

Example Prompts

Here are some example prompts you can try with the Frontend Agent:

1. **Todo Application:**

"Create a simple todo list application with add, complete, and delete functionality"

2. **Landing Page:**

"Design a modern landing page for a tech startup with hero section, features, and contact form"

3. **Contact Form:**

"Build a contact form with name, email, and message fields with validation"

4. **Dashboard:**

"Create a simple dashboard with cards showing statistics and a sidebar navigation"

5. **Portfolio Site:**

"Design a personal portfolio website with sections for about, projects, and contact"

Agent Capabilities

Core Functions

1. **generate_html_component(description, requirements)**

- Generates complete HTML applications based on text descriptions
- Returns structured response with HTML code and explanations
- Supports various types of web applications

2. **validate_and_optimize_code(html_code)**

- Validates HTML structure and best practices
- Provides optimization suggestions
- Checks for accessibility and performance considerations

Built-in Templates

The agent includes several pre-built templates:

- **Todo Application:** Full-featured task management app
- **Landing Page:** Modern business/startup landing page
- **Contact Form:** Form with validation and submission handling
- **Basic Template:** Customizable starting point for any application

Code Quality Standards

The Frontend Agent follows these standards:

- **HTML5 Semantic Structure:** Uses proper semantic elements
- **Responsive Design:** Mobile-first approach with Tailwind CSS
- **Accessibility:** WCAG guidelines and best practices
- **Modern JavaScript:** ES6+ features and clean code patterns
- **Performance:** Optimized loading and minimal dependencies
- **Documentation:** Well-commented code with explanations

Project Structure

```
frontend_agent/  
├── __init__.py      # Package initialization  
├── agent.py         # Main agent implementation  
├── .env            # Environment configuration  
├── README.md       # This documentation  
├── examples/       # Example generated applications  
│   ├── todo_app.html  
│   └── landing_page.html
```

Advanced Features

Multi-Agent Integration

The Frontend Agent is designed to work as part of a multi-agent system:

- Receives tasks from a Coordinator Agent
- Can collaborate with backend agents for full-stack applications
- Supports iterative development and feedback loops

Customization Options

You can customize the agent's behavior by:

- Modifying the instruction prompt in `agent.py`
- Adding new template functions
- Extending the validation rules
- Adding support for additional frameworks (when requested)

Troubleshooting

Common Issues

1. "Command 'adk' not found"

- Ensure virtual environment is activated
- Verify ADK installation: `pip show google-adk`
- Check Python version (3.9+ required)

2. Authentication Errors

- Verify API key in `.env` file
- Check internet connection
- Ensure API key has proper permissions

3. Agent Not Found in Dropdown

- Run `adk web` from parent directory of `frontend_agent`
- Check project structure matches requirements
- Verify `__init__.py` file exists

Getting Help

- [ADK Documentation](#)
- [ADK GitHub Repository](#)
- [Google AI Studio](#)

Contributing

To extend the Frontend Agent:

1. Add new template functions to `agent.py`
2. Update the instruction prompt for new capabilities
3. Add validation rules for new technologies
4. Create example applications in the `examples/` folder

License

This project follows the same license as Google's Agent Development Kit.

Example Interaction

User: "Create a simple todo list application"

Frontend Agent: "I'll create a modern todo list application with the following features:

- Add new tasks with an input field
- Mark tasks as complete/incomplete

- Delete unwanted tasks
- Show remaining task count
- Responsive design with Tailwind CSS

The application will be contained in a single HTML file with embedded CSS and JavaScript.

[Generates complete HTML application with all functionality]

Next Steps

After setting up the Frontend Agent, you can:

1. Try the example prompts above
2. Experiment with different types of applications
3. Integrate with other ADK agents for full-stack development
4. Deploy your generated applications to web hosting services
5. Extend the agent with additional capabilities