Building Your First Agent with TypeScript and React

SolveStation LABS

2025/10/25

SolveStation

CONTENTS

SolveStation

CONTENTS

SolveStation

LET'S MOVE ON

CHAPTER.01

Meet the Architect

SolveStation

Meet the Architect

Software Engineer & Poet

O Captain My Captain I architect distributed systems, cloud solutions, and AI-powered applications. Specialized in TypeScript, C#, Python, and enterprise-grade platforms including fintech and telehealth systems.

Passionate poet who believes elegant code mirrors poetry both demand precision, creativity, and soul. Creator of Rxsenty AI drug interaction checker.

Contact: awwalbalogun06@gmail.com
LinkedIn: linkedin.com/in/balogun14

SolveStation

CHAPTER.02

Foundation Choices

SolveStation

Why React & TypeScript?

A Foundation for Robust and Elegant Agents

React

- Component Reusability: Perfect for real-time chat/quiz interfaces.
- Virtual DOM: Ensures smooth, high-performance UIs.
- Vast Ecosystem: Accelerates development with proven libraries.

TypeScript

- Type Safety: Catches errors at compile time, not runtime.
- Superior DX: IntelliSense and easier refactoring. refactoring.
- Scalability: Self-documenting code that scales across teams.

"The most elegant code is that which can be read like prose."

SolveStation

CHAPTER.03

AI Essentials

SolveStation

Demystifying AI Jargon

From Fundamentals to Agents

AI
Machines performing human-like tasks.

→

ML
Systems learning from data patterns.

→

NLP
Understanding human language.

LLM (Large Language Model)
Trained on massive text datasets to predict the next word (e.g., GPT, Gemini).

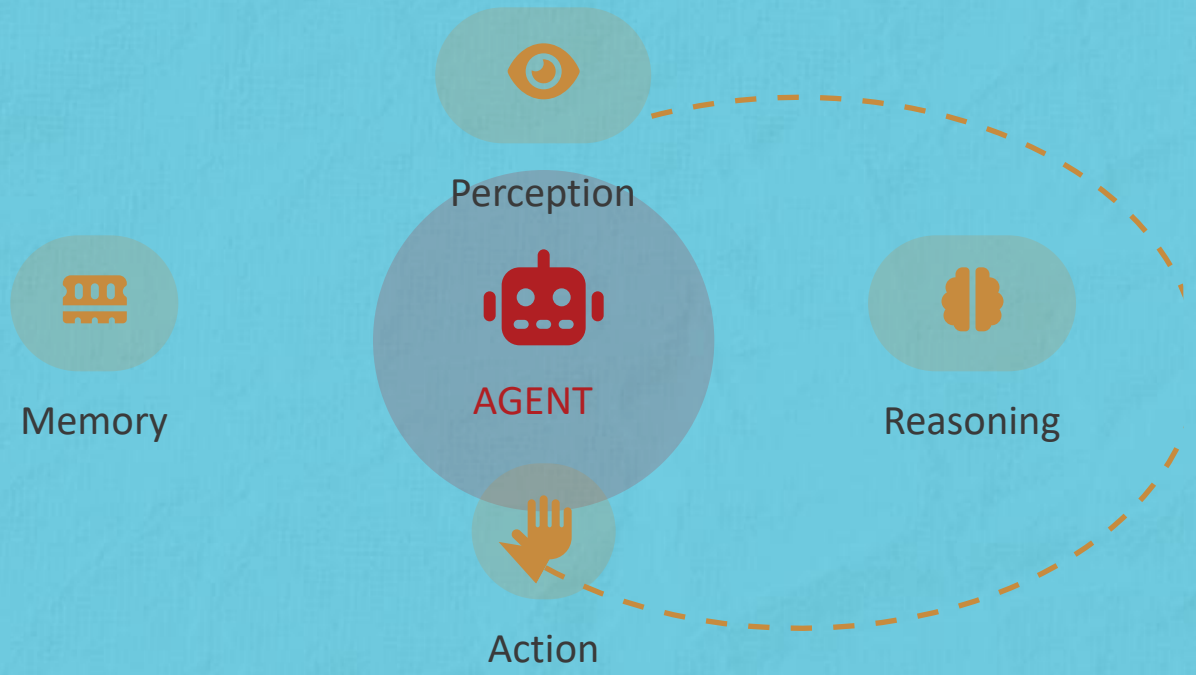
Agent
An autonomous system that perceives, reasons, and acts independently. independently.

"Every question is a seed of knowledge; let curiosity be your gardener."

SolveStation

Understanding AI Agents

The Cycle of Perception, Reasoning, Action & Memory



"An agent is you in the digital realm; it sees, thinks, and acts on your behalf."

SolveStation

CHAPTER.04

Agent Evolution

SolveStation

Our First Agent

A Simple CLI Chat Agent

Start with simplicity before embracing complexity. Build a basic conversational agent in minutes using:

- Node.js & TypeScript
- Google Gemini API

```
$ npm install @google/generative-ai
// 1. Initialize Gemini Client


// 2. Create Agent Executor


// 3. Define System Prompt


// 4. Run Agent Loop


> Hello! I'm ready to assist you.
```

SolveStation

Understanding LangGraph: Core Concepts

Element	Description	Analogy	TypeScript Relevance
State	A shared, central data structure that holds the current information and context for the entire application.	The **Whiteboard** (Shared Memory)	Defines the core data contract (e.g., a TypeScript interface) for the entire workflow.
Node	Individual functions or operations that perform specific tasks. Receives the state, processes it, and outputs the updated state.	An **Assembly Line Station** (Processing Unit)	Encapsulates single-responsibility functions that modify the state.
Graph	The overarching structure that maps out how nodes are connected and executed, defining the workflow.	A **Road Map** (Workflow Layout)	The blueprint that organizes all components.
Edges	Simple connections between nodes that determine the deterministic flow of execution.	A **Train Track** (Fixed Path)	Defines the sequential paths between nodes.
Conditional Edges	Specialized connections that decide the *next* node to execute based on a condition or logic applied to the current state.	A **Traffic Light** (Decision Point)	Implements crucial branching logic (e.g., if/else logic) within the graph.
Start/End Points	The virtual entry and exit points that mark where the workflow begins and concludes.	A Race's **Start and Finish Lines**	Defines the boundaries of the execution.
Tools & Tool Nodes	**Tools** are external functions (e.g., APIs, calculators) that nodes can call. **Tool Nodes** are the mechanisms that execute these tools.	A **Toolbox** (External Capabilities)	Enables the LLM to access external logic beyond its training data.
State Graph & Runnable	The **State Graph** manages the nodes, edges, and state to compile the workflow. **Runnables** are standardized, executable building blocks.	**Lego Bricks** (Modular Components)	Essential for compiling the complex graph into an executable unit.
Message Types	Different types of messages used for communication with LLMs, including Human, AI, System, and Tool messages.	**Labels** (Context Tags)	Crucial for maintaining structured conversation history and agent prompts.

"The best software solves real human problems."

SolveStation

From CLI to StudyBuddy

The Evolution of an Idea into a Solution

Simple CLI Chat

Multi-Tool Agent

StudyBuddy App

Problem
Study plan confusion & scattered resources.

Solution
AI-powered organization & engagement.

Result
Structured learning & exam confidence.

"The best software solves real human problems."

SolveStation

LET'S MOVE ON

CHAPTER.05

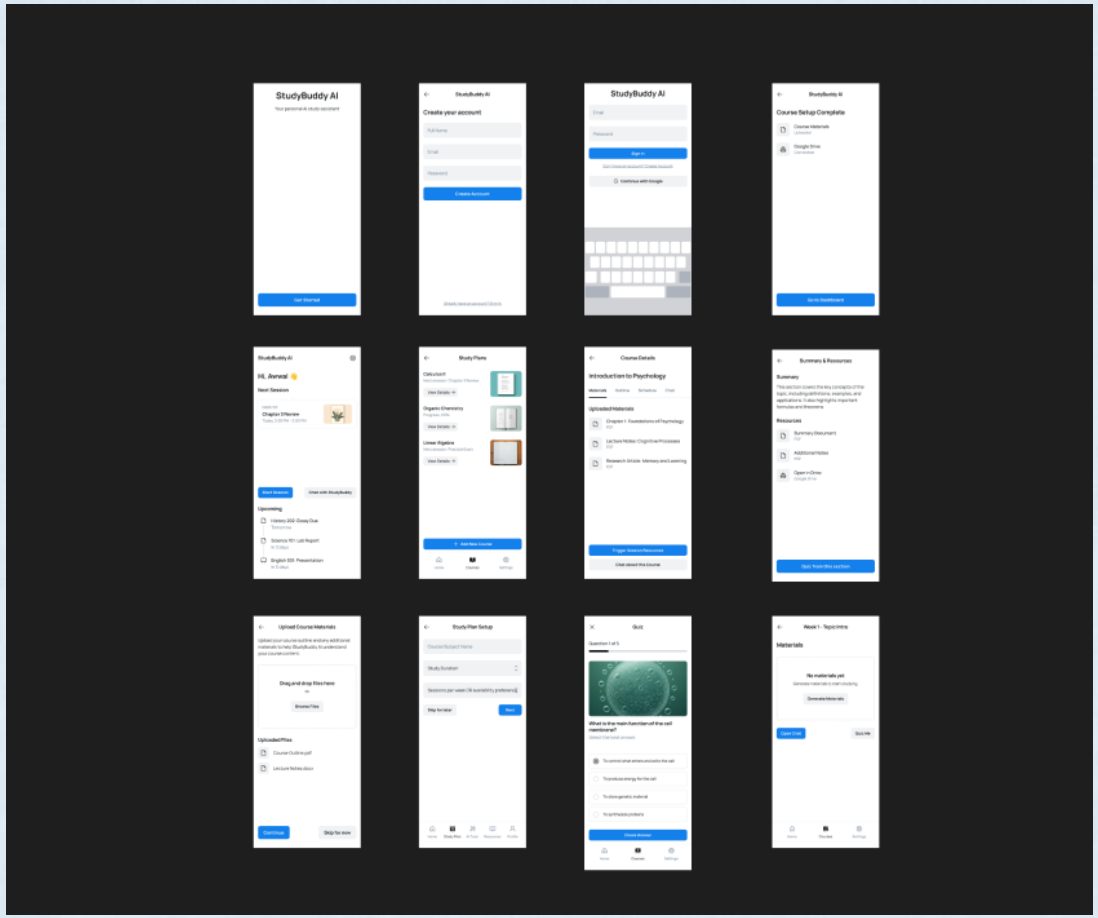
StudyBuddy Vision

SolveStation

StudyBuddy Vision

Your Personal AI Tutor

Transforming chaotic learning into structured, engaging conversations. The core promise is simple: Organized Learning → Better Retention → Exam Confidence.



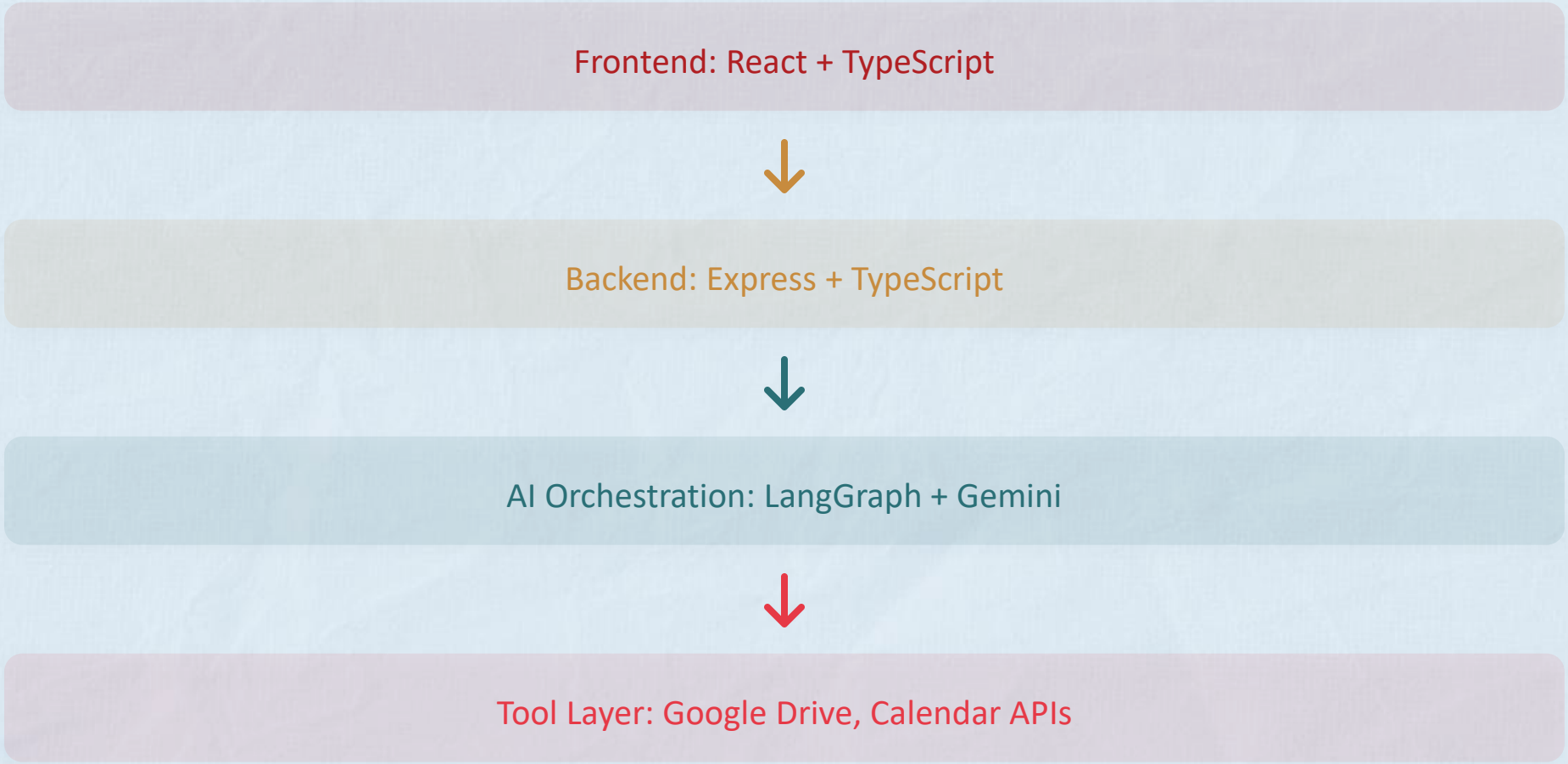
- Upload materials for AI analysis.
- Generate concise explanations & resources.
- Auto-schedule study sessions.
- Engage with interactive quizzes.

SolveStation

"Learning is a conversation, not a monologue."

System Architecture Deep Dive

A Clean Separation of Concerns



"Good architecture is like a well-organized library; everything is where you expect it."

SolveStation

CHAPTER.06

Multi-Agent Design

SolveStation

Why Multiple Agents?

A single agent buckles under too many responsibilities, becoming a tangled mess that's hard to debug and scale. Specialization is the key to mastery.

> **Benefits of Specialization:**
>
> Modularity, Testability, Scalability, Maintainability, Clear Responsibility.

"Divide and conquer; complex problems yield to focused minds."

Research Agent

Analyzes materials

Content Agent

Creates study guides

Scheduling Agent

Manages calendar

Quiz Agent

Generates questions

SolveStation

Backend Walkthrough

The backend is the spine of your application; it must be both strong and flexible. Our Express + TypeScript server is the central hub for orchestration and integration.

Core Responsibilities:

API endpoints, Agent orchestration, Google API integrations, Error handling.

Key Patterns:

Middleware, Service Layer, Repository Pattern, Environment Config.

SolveStation

CHAPTER.07

Live Implementation

SolveStation

Live Coding Session

Building a Simple Agent from Scratch

"Code is poetry written in logic; let your thoughts flow naturally."

SolveStation

Demo: StudyBuddy in Action

From Upload to Organized Learning

1. Student Uploads Material

A biology chapter is uploaded.

2. Agents Orchestrate

Analysis, content creation, scheduling.

3. Results Delivered

Guides & quizzes saved to Drive.

"See the magic when your ideas become reality."

SolveStation

CHAPTER.08

Achievement Recap

SolveStation

What We've Built

- A multi-agent system with clear responsibilities.
- Seamless integration with Google APIs.
- Robust frontend-backend communication.
- Production-ready architectural patterns.

"What we've built today is tomorrow's foundation."

Key Lessons Learned

- Start simple, then scale complexity.
- Tools are your agent's superpowers.
- Orchestration is critical for coordination.
- Type safety pays dividends in reliability.

SolveStation

Next Steps & Beyond

Immediate Next Steps

- Run & experiment locally
- Modify prompts & observe
- Add custom tools
- Deploy to production

Advanced Topics

- Fine-tuning LLMs
- Multi-modal agents
- Agent evaluation
- Building frameworks

Resources

- LangChain & Gemini Docs
- React & Express Guides
- StudyBuddy Repo
- AI/ML Beginner Guides

"This is just the beginning; the agents you'll build will astound you."

SolveStation

CHAPTER.09

Connection & Closure

SolveStation

Questions?

Discussion & Community Engagement

"The best questions are those asked with curiosity, not certainty."

SolveStation

Muhammed-Awwal Balogun
awwalbalogun06@gmail.com

Essential Resources & Links

LangChain Docs
js.langchain.com

Google Gemini API
cloud.google.com/vertex-ai

React + TypeScript
react.dev/learn/typescript

Express + TypeScript
expressjs.com

SolveStation

How to Reach Me

awwalbalogun06@gmail.com

linkedin.com/in/balogun14

If e reach your turn put your phone number :lol

Let's Connect!

Find me on LinkedIn

Email us your wildest Ideas we will build it at hello@solvestation.tech

Let's build something that makes the internet slightly better.

SolveStation

Build Great Things.

Share Them With The World.

We stand at the intersection where intelligent systems meet clear architecture to solve real problems, together as a community. Every agent you build carries a piece of your unique vision—make it count.

"The path from code to curiosity leads directly to innovation. You've walked this path today; now run with it." run with it."

SolveStation

CONTACT US

hello@solvestation.tech

solvestationltd@gmail.com

SolveStation