

JurneeGo – Founding Platform & AI Systems Engineer

JurneeGo

7 May, 2026

Table of Contents

1	About JurneeGo / Giới Thiệu Về JurneeGo.....	5
2	Role Overview / Mô Tả Vai Trò	6
3	What You Will Build / Những Gì Bạn Sẽ Xây Dựng	7
3.1	Core Systems Architecture / Kiến Trúc Hệ Thống.....	7
3.2	AI Systems (AI-Focused Engineers)	7
3.3	Platform & DevOps (Platform-Focused Engineers)	8
4	Engineering Governance / Quản Trị Kỹ Thuật.....	9
5	Ideal Profile / Hồ Sơ Ứng Viên Lý Tưởng	10
6	Language Requirement / Yêu Cầu Ngôn Ngữ.....	11
7	What Success Looks Like (First Phase).....	12
8	Compensation & Structure / Quyền Lợi.....	13
9	How to Apply / Cách Ứng Tuyển	14
9.1	Engineering Thought Exercise / Bài Tập Tư Duy Kỹ Thuật	14
10	JurneeGo's Founding Engineer #2: The System Stabilizer & Scale Architect.....	16
10.1	🧠 JurneeGo – Ideal Founding Engineer #2 Profile	16
10.1.1	🎯 Core Role Definition	16
10.2	I. 🧬 Role Mission (Non-Negotiable)	16
10.3	II. 🧠 Archetype (Who This Person Is).....	17
10.3.1	🧱 “Builder–Operator Hybrid”	17
10.4	III. 🔥 Core Competencies.....	17
10.4.1	1. ⚙️ Backend Systems Ownership (CRITICAL)	17
10.4.2	2. 🧠 AI Routing + Cost Optimization (CORE IP)	17
10.4.3	3. 🛡️ Safety & Compliance Engineering (MANDATORY)	18
10.4.4	4. 📡 Real-Time Systems (IMPORTANT)	18
10.4.5	5. 💰 Cost Engineering Mindset	19
10.4.6	6. 🔍 Observability & Debugging	19
10.5	IV. 🚫 What They Are NOT	19
10.5.1	❌ Pure Frontend Engineer	19
10.5.2	❌ Enterprise DevOps Only	19

10.5.3	✗	Research AI Engineer	19
10.5.4	✗	“Prompt Engineer”	20
10.6	V. 🧠	Personality Traits (VERY IMPORTANT).....	20
10.6.1	1.	Low Ego, High Ownership	20
10.6.2	2.	Calm Under Chaos	20
10.6.3	3.	Obsession with Tradeoffs	20
10.6.4	4.	Builder Mentality	20
10.7	VI. ✂	Interaction Model (Vy + FE #2)	20
10.7.1		Vy (FE #1)	20
10.7.2		FE #2.....	21
10.8	VII. 📅	30-60-90 Day Expectations	21
10.8.1	🟢	First 30 Days.....	21
10.8.2	🟡	60 Days	21
10.8.3	🔴	90 Days	21
10.9	VIII. 🗣️	Interview Questions (Use These).....	22
10.9.1	🔥	System Thinking.....	22
10.9.2	🔥	AI Routing	22
10.9.3	🔥	Failure Mode.....	22
10.9.4	🔥	Real-Time Systems	22
10.9.5	🔥	Tradeoffs	23
10.10	IX. 🌟	Hiring Bar (Non-Negotiable).....	23
10.11	X. 💬	Offer Narrative (What You Tell Them)	23
10.12	🧠	Final Insight	24

Core Systems Architecture • Multi-Tenant Infrastructure • AI Safety Systems

English	Tiếng Việt
Location: Vietnam (Hybrid or Remote within Vietnam preferred)	Địa điểm: Việt Nam (Hybrid hoặc Remote trong Việt Nam)
Employment Type: Full-time	Hình thức: Toàn thời gian
Company: JurneeGo	Công ty: JurneeGo



1 About JurneeGo / Giới Thiệu Về JurneeGo

English	Tiếng Việt
JurneeGo is building a safe AI learning platform for children that connects children, parents, and teachers inside a transparent and governed AI environment.	JurneeGo đang xây dựng nền tảng AI học tập an toàn cho trẻ em , kết nối trẻ em, phụ huynh và giáo viên trong một môi trường AI minh bạch và có kiểm soát.
Most AI systems today were not designed for children .	Phần lớn các hệ thống AI hiện nay không được thiết kế dành cho trẻ em .
Children are either locked out entirely, or exposed to tools that were never designed for learning environments.	Trẻ em hoặc bị loại khỏi các nền tảng AI , hoặc tiếp xúc với các công cụ không phù hợp với môi trường học tập.
JurneeGo solves this by combining:	JurneeGo giải quyết vấn đề này bằng cách kết hợp:
<ul style="list-style-type: none"> • Child-safe AI interaction 	<ul style="list-style-type: none"> • AI an toàn và phù hợp với trẻ em
<ul style="list-style-type: none"> • Parent visibility into learning journeys 	<ul style="list-style-type: none"> • Phụ huynh có thể theo dõi hành trình học tập của con
<ul style="list-style-type: none"> • Teacher integration with classroom learning 	<ul style="list-style-type: none"> • Tích hợp với giáo viên và chương trình học
<ul style="list-style-type: none"> • Compliance with global child data privacy laws 	<ul style="list-style-type: none"> • Tuân thủ các quy định bảo vệ dữ liệu trẻ em trên toàn cầu
Our mission is to build the technical infrastructure that makes AI safe and usable in education environments .	Sứ mệnh của chúng tôi là xây dựng hạ tầng kỹ thuật giúp AI có thể được sử dụng an toàn trong giáo dục .

2 Role Overview / Mô Tả Vai Trò

English	Tiếng Việt
We are hiring a Founding Engineer to help build the core technical foundation of the JurneeGo platform.	Chúng tôi đang tìm Founding Engineer để cùng xây dựng nền tảng kỹ thuật cốt lõi của JurneeGo.
This is a system ownership role , not just feature development.	Đây là vị trí sở hữu hệ thống , không chỉ phát triển tính năng.
You will work directly with the Head of Engineering to design and implement core backend systems, AI orchestration layers, safety enforcement logic, and platform reliability foundations.	Bạn sẽ làm việc trực tiếp với Head of Engineering để thiết kế và xây dựng backend, hệ thống orchestration cho AI, các lớp bảo vệ an toàn và nền tảng vận hành ổn định.
We are looking for engineers who think in systems and architecture , not just tasks.	Chúng tôi tìm kiếm những kỹ sư có tư duy hệ thống và kiến trúc , không chỉ làm việc theo task.
You may be strongest in either:	Ứng viên có thể mạnh về một trong hai hướng:
<ul style="list-style-type: none"> • AI systems engineering 	<ul style="list-style-type: none"> • Hệ thống AI
<ul style="list-style-type: none"> • Platform / DevOps engineering 	<ul style="list-style-type: none"> • Platform / DevOps
Both profiles are valuable. What matters most is system-level thinking and ownership .	Cả hai hướng đều phù hợp. Điều quan trọng nhất là tư duy hệ thống và tinh thần ownership .

3 What You Will Build /Những Gì Bạn Sẽ Xây Dựng

3.1 Core Systems Architecture / Kiến Trúc Hệ Thống

English	Tiếng Việt
Design scalable backend architecture (domain models, APIs, state management).	Thiết kế backend có khả năng mở rộng (domain model, API, state management).
Build secure multi-tenant services with strict boundary enforcement.	Xây dựng hệ thống multi-tenant an toàn với phân quyền rõ ràng.
Implement governance logic and permission systems.	Phát triển hệ thống governance và permission.
Maintain high-integrity audit logging and data visibility layers.	Triển khai audit log và hệ thống giám sát dữ liệu.

3.2 AI Systems (AI-Focused Engineers)

English	Tiếng Việt
Implement AI orchestration pipelines.	Xây dựng pipeline orchestration cho AI.
Design prompt packaging and context management systems.	Thiết kế hệ thống prompt và context management.
Implement guardrails and safety enforcement logic.	Phát triển guardrails và cơ chế bảo vệ an toàn.
Build evaluation and monitoring pipelines for AI responses.	Xây dựng hệ thống đánh giá và theo dõi phản hồi AI.
Optimize reliability and cost of AI usage.	Tối ưu độ ổn định và chi phí sử dụng AI.

3.3 Platform & DevOps (Platform-Focused Engineers)

English	Tiếng Việt
Design CI/CD pipelines and release processes.	Thiết kế pipeline CI/CD và quy trình release.
Implement infrastructure-as-code and environment management.	Quản lý hạ tầng bằng infrastructure-as-code.
Build observability systems (logs, metrics, tracing).	Xây dựng hệ thống monitoring (logs, metrics, tracing).
Improve reliability, deployment safety, and cloud cost governance.	Nâng cao độ ổn định hệ thống và quản lý chi phí cloud.

4 Engineering Governance / Quản Trị Kỹ Thuật

English	Tiếng Việt
Review vendor contributions touching core systems.	Review code và đóng góp từ vendor liên quan đến hệ thống lõi.
Define and enforce coding and architecture standards.	Thiết lập và duy trì tiêu chuẩn coding và kiến trúc hệ thống.
Participate in release gating and risk review processes.	Tham gia đánh giá rủi ro trước khi phát hành.
Contribute to system reliability and scaling strategy.	Hỗ trợ chiến lược mở rộng và ổn định hệ thống.

5 Ideal Profile / Hồ Sơ Ứng Viên Lý Tưởng

English	Tiếng Việt
Strong system-level thinker, not only feature-focused.	Tư duy hệ thống tốt, không chỉ tập trung vào feature.
Experience designing backend or distributed systems.	Có kinh nghiệm backend hoặc distributed systems.
Comfortable designing multi-tenant systems and RBAC models.	Hiểu về multi-tenant architecture và RBAC.
Experience working with cloud environments (AWS preferred).	Có kinh nghiệm cloud (AWS là lợi thế).
Deep strength in AI systems OR Platform/DevOps.	Mạnh về AI systems hoặc Platform/DevOps.
High ownership mindset and comfortable operating with ambiguity.	Tinh thần ownership cao và chủ động trong môi trường startup.
AI-native engineer using modern AI coding tools effectively.	Quen thuộc với các công cụ AI hỗ trợ lập trình.

6 Language Requirement / Yêu Cầu Ngôn Ngữ

English	Tiếng Việt
English is helpful but not required.	Tiếng Anh là lợi thế nhưng không bắt buộc.
Technical capability and system thinking are the primary evaluation criteria.	Khả năng kỹ thuật và tư duy hệ thống là yếu tố đánh giá chính.

7 What Success Looks Like (First Phase)

English	Tiếng Việt
Core backend foundation established and hardened.	Backend cốt lõi được xây dựng và ổn định.
AI orchestration or platform reliability structured clearly and safely.	Hệ thống AI orchestration hoặc platform reliability hoạt động rõ ràng và an toàn.
Safety enforcement system implemented with regression coverage.	Hệ thống guardrails và bảo vệ an toàn được triển khai đầy đủ.
Vendor systems integrated cleanly with internal platform architecture.	Các hệ thống vendor tích hợp tốt với core platform.
Deployment pipelines and platform reliability stabilized.	Pipeline deployment và reliability được thiết lập ổn định.
Core platform knowledge not dependent on a single individual.	Kiến thức hệ thống không phụ thuộc vào một cá nhân duy nhất.

8 Compensation & Structure / Quyền Lợi

English	Tiếng Việt
Competitive salary based on experience.	Lương cạnh tranh theo năng lực.
Meaningful equity participation. 0.5% to 1.5%	Cổ phần công ty (equity). 0.5% to 1.5%
Founding-level influence on system architecture.	Vai trò có ảnh hưởng lớn trong kiến trúc hệ thống.
Direct collaboration with Head of Engineering.	Làm việc trực tiếp với Head of Engineering.
Opportunity to shape long-term technical direction.	Cơ hội định hình hướng phát triển kỹ thuật lâu dài.
Flexible hybrid work	Chế độ làm việc kết hợp linh hoạt
Paid annual leave	Nghỉ phép năm có lương
Professional Learning & Growth <ul style="list-style-type: none"> Free access to selected courses from American University in Vietnam (AUV) Access to academic research databases and university libraries Education-discount access to developer tools and cloud platforms 	Học tập & Phát triển nghề nghiệp <ul style="list-style-type: none"> Truy cập miễn phí vào các khóa học chọn lọc từ Đại học Hoa Kỳ tại Việt Nam (AUV) Truy cập vào các cơ sở dữ liệu nghiên cứu học thuật và thư viện đại học Giảm giá dành cho sinh viên khi sử dụng các công cụ phát triển và nền tảng đám mây
Engineering Tools <ul style="list-style-type: none"> Modern AI-assisted development tools Cloud experimentation budget Education licenses for developer software 	Công cụ kỹ thuật <ul style="list-style-type: none"> Các công cụ phát triển hiện đại hỗ trợ AI Ngân sách thử nghiệm trên nền tảng đám mây Giấy phép phần mềm dành cho sinh viên
Environment <ul style="list-style-type: none"> Work alongside experienced startup founders and international educators Opportunity to help build the world's first safe AI platform for children 	Môi trường làm việc <ul style="list-style-type: none"> Làm việc cùng với những người sáng lập công ty khởi nghiệp giàu kinh nghiệm và các nhà giáo dục quốc tế Cơ hội tham gia xây dựng nền tảng AI an toàn đầu tiên trên thế giới dành cho trẻ em

9 How to Apply / Cách Ứng Tuyển

English	Tiếng Việt
Please submit your CV or LinkedIn profile along with a short introduction.	Vui lòng gửi CV hoặc LinkedIn profile kèm một đoạn giới thiệu ngắn.
We encourage candidates to include links to systems they have built or owned. This could include GitHub repositories, production systems, infrastructure projects, or technical write-ups describing systems you helped design.	Chúng tôi khuyến khích ứng viên gửi kèm các liên kết đến những hệ thống mà bạn đã trực tiếp xây dựng hoặc sở hữu. Có thể bao gồm GitHub repository, hệ thống production, dự án hạ tầng, hoặc bài viết kỹ thuật mô tả các hệ thống bạn đã tham gia thiết kế.
Contact: careers@jurneego.com ¹	Liên hệ: careers@jurneego.com ²

9.1 Engineering Thought Exercise / Bài Tập Tư Duy Kỹ Thuật

English	Tiếng Việt
Optional Engineering Thought Exercise	Bài Tập Tư Duy Kỹ Thuật (Không Bắt Buộc)
To help us understand how you think about systems, we invite interested candidates to answer the question below.	Để giúp chúng tôi hiểu cách bạn tư duy về hệ thống, chúng tôi mời các ứng viên quan tâm trả lời câu hỏi dưới đây.
This is not a formal test and there is no single correct answer. We are interested in your reasoning and architecture thinking.	Đây không phải bài kiểm tra và không có đáp án duy nhất. Chúng tôi quan tâm đến cách bạn suy nghĩ và thiết kế hệ thống.
Question:	Câu hỏi:
<i>Describe the architecture you would design to build a safe multi-tenant AI platform used by children, parents, and teachers.</i>	<i>Hãy mô tả kiến trúc hệ thống bạn sẽ thiết kế để xây dựng một nền tảng AI multi-tenant an toàn dành cho trẻ em, phụ huynh và giáo viên.</i>
Your design should consider:	Thiết kế của bạn nên xem xét các yếu tố sau:
<ul style="list-style-type: none"> Multi-tenant system boundaries 	<ul style="list-style-type: none"> Phân tách hệ thống multi-tenant

1. <mailto:careers@jurneego.com>

2. <mailto:careers@jurneego.com>

English	Tiếng Việt
<ul style="list-style-type: none"> • Role-based access control (child / parent / teacher) 	<ul style="list-style-type: none"> • Phân quyền theo vai trò (trẻ em / phụ huynh / giáo viên)
<ul style="list-style-type: none"> • Safety guardrails and misuse prevention 	<ul style="list-style-type: none"> • Cơ chế guardrails và phòng chống lạm dụng
<ul style="list-style-type: none"> • AI orchestration and prompt context management 	<ul style="list-style-type: none"> • Hệ thống orchestration cho AI và quản lý context
<ul style="list-style-type: none"> • Logging, monitoring, and audit visibility 	<ul style="list-style-type: none"> • Logging, monitoring và khả năng audit
<ul style="list-style-type: none"> • Reliability and deployment strategy 	<ul style="list-style-type: none"> • Chiến lược reliability và triển khai
You may answer with:	Bạn có thể trả lời bằng:
<ul style="list-style-type: none"> • A short written explanation 	<ul style="list-style-type: none"> • Một đoạn mô tả ngắn
<ul style="list-style-type: none"> • A diagram or architecture sketch 	<ul style="list-style-type: none"> • Sơ đồ kiến trúc hệ thống
<ul style="list-style-type: none"> • A system breakdown of components 	<ul style="list-style-type: none"> • Phân tích các thành phần hệ thống
We are primarily evaluating:	Chúng tôi đánh giá chủ yếu dựa trên:
<ul style="list-style-type: none"> • System thinking 	<ul style="list-style-type: none"> • Tư duy hệ thống
<ul style="list-style-type: none"> • Safety awareness 	<ul style="list-style-type: none"> • Nhận thức về an toàn hệ thống
<ul style="list-style-type: none"> • Clarity of architecture 	<ul style="list-style-type: none"> • Sự rõ ràng trong thiết kế kiến trúc
Submissions are optional but often helpful for starting deeper technical conversations.	Phần trả lời là không bắt buộc nhưng thường giúp mở đầu các cuộc trao đổi kỹ thuật sâu hơn.

[JurneeGo: A Unique Opportunity for Real Ownership and Wealth Creation in Startups](https://jurneego.atlassian.net/wiki/spaces/JurneeGo/pages/11436035/JurneeGo+A+Unique+Opportunity+for+Real+Ownership+and+Wealth+Creation+in+Startups)³

3. <https://jurneego.atlassian.net/wiki/spaces/JurneeGo/pages/11436035/JurneeGo+A+Unique+Opportunity+for+Real+Ownership+and+Wealth+Creation+in+Startups>

10 JurneeGo's Founding Engineer #2: The System Stabilizer & Scale Architect

10.1 🧠 JurneeGo – Ideal Founding Engineer #2 Profile

10.1.1 🎯 Core Role Definition

"The System Stabilizer & Scale Architect"

If Vy is:

- ⚡ Speed
- ⚡ Product build

Then FE #2 is:

- 🧱 System integrity
- 🧱 Scalability
- 🧱 Cost control
- 🧱 Safety enforcement

10.2 I. 🧬 Role Mission (Non-Negotiable)

Primary Objective (first 90 days):

Take what Vy builds and make it:

- Reliable
- Scalable
- Safe
- Cost-efficient

Without slowing product velocity.

10.3 II. 🧠 Archetype (Who This Person Is)

You are NOT hiring:

- another frontend builder
- another “AI tinkerer”

You ARE hiring:

10.3.1 🧱 “Builder–Operator Hybrid”

Someone who:

- Has built systems AND run them in production
 - Thinks in failure modes, not just features
 - Understands tradeoffs (cost vs latency vs safety)
-

10.4 III. 🔥 Core Competencies

10.4.1 1. ⚙️ Backend Systems Ownership (CRITICAL)

Must be able to own:

- APIs
- Data flow
- Event systems (WebSockets / real-time)
- Session architecture (SSA)

From your system:

- Child → AI → Parent → Teacher loop

👉 This is NOT trivial. This is your core product.

10.4.2 2. 🧠 AI Routing + Cost Optimization (CORE IP)

They MUST understand:

- Multi-model routing
- Token cost dynamics
- Latency tradeoffs

Example:

- Cheap model → 80% traffic

- Premium model → 20%
- Maintain UX consistency

From your architecture:

- A/B model routing
- Cost-aware scaling

👉 If they don't get this → reject.

10.4.3 3. Safety & Compliance Engineering (MANDATORY)

They must think:

"What happens if this goes wrong with a child?"

Must understand:

- Input/output filtering
- Guardrails (pre + post LLM)
- Logging + audit trails
- Crisis escalation flows

From your system:

- Safety layer BEFORE model
- Safety layer AFTER model

👉 This is NOT optional. This is your moat.

10.4.4 4. Real-Time Systems (IMPORTANT)

Your product requires:

- Live parent monitoring
- Shared sessions
- Multi-user sync

They should understand:

- WebSockets
- Pub/Sub (Redis)
- Event-driven architecture

10.4.5 5. Cost Engineering Mindset

They think like this:

“Every feature = cost per query”

Must be able to:

- Model cost per user
 - Optimize infra vs LLM spend
 - Implement caching strategies
-

10.4.6 6. Observability & Debugging

They must build:

- Logging systems
- Monitoring dashboards
- Failure tracing

Because:

- You will have AI unpredictability
 - You will have safety edge cases
-

10.5 IV. What They Are NOT

Reject if they are:

10.5.1 Pure Frontend Engineer

No.

10.5.2 Enterprise DevOps Only

Too slow, too rigid.

10.5.3 Research AI Engineer

Too theoretical.

10.5.4 ❌ “Prompt Engineer”

Useless at this stage.

10.6 V. 🧠 Personality Traits (VERY IMPORTANT)

10.6.1 1. Low Ego, High Ownership

- Works WITH Vy, not against
- Doesn't create silos

10.6.2 2. Calm Under Chaos

- Early startup = messy
- Must handle ambiguity

10.6.3 3. Obsession with Tradeoffs

- Not “best solution”
- “right solution now”

10.6.4 4. Builder Mentality

- Ships
 - Fixes
 - Iterates
-

10.7 VI. ✂️ Interaction Model (Vy + FE #2)

This is critical.

10.7.1 Vy (FE #1)

- Moves fast
- Builds features
- Experiments

10.7.2 FE #2

- Hardens systems
- Refactors intelligently
- Prevents future collapse

👉 Together:

- Speed + Stability
-

10.8 VII. 30-60-90 Day Expectations

10.8.1 First 30 Days

Goal: Understand system deeply

- Review architecture
- Map data flow
- Identify risks

Deliverable:

👉 “System Risk Map”

10.8.2 60 Days

Goal: Own infrastructure layer

- Improve API structure
- Optimize AI routing
- Implement logging + monitoring

Deliverable:

👉 Stable backend system

10.8.3 90 Days

Goal: Become system co-architect

- Improve performance
- Reduce costs
- Lead backend decisions

Deliverable:

👉 Scalable production-ready system

10.9 VIII. 🧪 Interview Questions (Use These)

10.9.1 🔥 System Thinking

“Design how a child query flows through JurneeGo.”

Look for:

- safety layer first
 - routing logic
 - logging
-

10.9.2 🔥 AI Routing

“How do you reduce LLM cost by 50% without hurting UX?”

10.9.3 🔥 Failure Mode

“What happens if the AI gives a dangerous answer?”

10.9.4 🔥 Real-Time Systems

“How would you build parent live monitoring?”

10.9.5 🔥 Tradeoffs

“Would you prioritize latency or cost? Why?”

10.10 IX. 🚀 Hiring Bar (Non-Negotiable)

They must:

- ✓ Think in systems
- ✓ Understand AI infra
- ✓ Care about safety
- ✓ Be cost-aware
- ✓ Ship fast

If any missing → pass.

10.11 X. 💬 Offer Narrative (What You Tell Them)

You're not hiring an employee. You're recruiting a builder.

Subject: Join JurneeGo as Founding Engineer #2

You won't just write code here.

You'll define how a system that children depend on actually works.

Every decision you make will affect:

- how safe the product is
- how fast it scales
- how much it costs to run

You'll take what we build—and make it real, stable, and world-ready.

If that excites you, you'll do the best work of your career here.

– Gui

10.12 Final Insight

Your biggest advantage right now is:

You are building BOTH a product AND an architecture moat.

FE #1 builds the product.
FE #2 protects the moat.