📢 Day 7: Share Results

It's our last day of the crash course! We have a data processing pipeline and an agent that's already tested and deployed.

But is it ready to showcase?

In today's lesson, we'll refine our code repository and transform it into a compelling portfolio project. Not only will you be able to share it on social media, but also with hiring managers when applying for AI jobs.

Our plan for today:

- Write a project README
- Create demo videos and screenshots
- Share the project and get feedback

At the end of the lesson, you'll have a project that you can add to your CV.

Let's start with the most important file in your GitHub repository: README.md

1. Project README.md

When I review portfolio projects or those created by our students, that's the first thing I look at. And often, it's the only thing I look at - unless the project is exciting and I want to learn about some implementation details (which doesn't happen often).

So, when I'm hiring, I don't have a lot of time to go through the project's code. That is why it's very important to make sure your README is as good as possible.

For my personal projects, I don't have a well-defined formula for READMEs. I usually try to provide sufficient information about the project's purpose at the beginning, then demonstrate how to set it up, and finally show how to use it.

But I asked ChatGPT to give me the formula, and this is the answer:

1. Project Title & Description
   - Short, descriptive title
   - One-liner explaining the purpose of the project
   - Optional: badges (build status, license, version, etc.)
2. Overview
   - Explain the problem your project solves

   - Why it's useful / unique
   - Screenshots, GIFs, videos or diagrams (if applicable)
3. Installation
   - Step-by-step instructions to set up locally
   - Requirements (dependencies, versions, platforms)
   - Code snippets for installation
4. Usage
   - Examples of how to run or use the project
   - Command line examples, API calls, or screenshots
   - Configuration options (if relevant)
5. Features
   - Highlight key features
   - Roadmap or upcoming features (optional)
6. Contributing
   - Guidelines for pull requests
   - Link to `CONTRIBUTING.md` if you have one
   - Mention coding standards / style guide
7. Tests
   - How to run tests
   - Example commands for unit/integration testing
8. Deployment (optional)
   - Steps for deploying to production
   - Any CI/CD notes
9. FAQ / Troubleshooting (optional)
   - Common issues and fixes
10. Credits / Acknowledgments
   - Contributors, libraries, tutorials, inspirations
11. License
   - License type (e.g., MIT, Apache 2.0)
   - Link to `LICENSE` file

I really liked this answer, so I copied it here without editing. We may skip some of them, like contributing or licensing, but it's a very good plan.

As a big fan of AI-assisted development, I'll naturally use an AI tool to help me write the README file.

Here's what I'll do:

- Ask ChatGPT to create me a project README
- Copy the plan in the prompt so it knows which structure to follow
- Dictate the project idea (I'll just use my microphone and its voice recognition feature)
- Upload all the project files
- Get the result and polish it

You can see my interaction with ChatGPT here.

Once I got the README file, I spent some time editing it. You can see the final result here: https://github.com/alexeygrigorev/aihero/tree/main/code

I removed a few sections and added Evaluations.

2. Evaluations

In my case, all the code related to the evaluation was in the main (and only) Jupyter notebook. So I thought it's a good idea to clean it and move this part of the code elsewhere, like we did yesterday with the main agent code.

If it's the same case for you, this is what you can do:

- Put the code for generating questions in `eval/data-gen.ipynb` (or create a Python script)
- Put the evaluation logic in `eval/evaluations.ipynb` (or also create a script)

Then we add the "Evaluation" section in the README. While the README structure ChatGPT suggested is great, it's for "usual" software projects. In our case, it's AI, so we also need to mention evaluations, as they are a very important part of our project.

It doesn't really matter how exactly you organize your evaluation code. The most important part right now is mentioning that:

- You have an evaluation
- What are the numbers

You can see my "Evaluations" section here.

3. Demo

Next, we can record a short video and place it in the README. Also, add gifs in a few places.

Here's mine:

- The main demo video showing the agent in action
- CLI gif demonstrating command-line usage
- Streamlit gif showing the web interface

It could be just screenshots, without demos/gifs. Here it's up to you. The important part is to have some visuals.

It really helps to see the results visually. When I review someone's repository, I won't have time to run the code and see how it works. So having some visuals will undoubtedly help.

4. Sharing the project

Now that we have polished our repository, we're ready to share the project with the world!

This is your moment to celebrate what you've built over the past week.

You've created a complete AI agent system from scratch - from data ingestion to deployment. That's something to be proud of!

So, share the results. Focus on:

- What problem your agent solves
- The technical skills you demonstrated
- The results and evaluation metrics
- What you learned during the process

Remember, this isn't just about showing off. It's about connecting with the AI and data science community, getting feedback, and potentially opening doors for career opportunities.

Homework

- Polish your GitHub repository with a comprehensive README
- Organize your evaluation code into separate files
- Create demo videos, gifs, or screenshots
- Share your project on social media
- Celebrate completing the course!

Example post for LinkedIn

🎉 I just completed a 7-day AI Agents crash course.

Over the past week, I created a complete AI agent system from scratch:

✅ Built a data pipeline to process GitHub repositories
✅ Created a search engine with document indexing
✅ Developed an AI agent using Pydantic AI
✅ Implemented comprehensive evaluation system
✅ Built a web interface with Streamlit

My agent helps users search through [YOUR PROJECT DESCRIPTION] and provides accurate answers with source references.

📁 GitHub repo: [YOUR_REPO_LINK]

Huge thanks to @Alexey Grigorev for this incredible course. The hands-on approach made complex concepts accessible.

Who else is building AI agents? I'd love to see your projects!

Course link: https://alexeygrigorev.com/aihero/

You can also include the video you recorded.

Example post for Twitter/X

🎉 7-day AI Agent crash course: COMPLETE!

✅ Data pipeline
✅ Search engine
✅ AI agent
✅ Evals
✅ Web interface

From idea → agent in 1 week! 🚀

📁 Code: [YOUR_REPO_LINK]

Thanks @AI_Grigor for an amazing course!

Course: https://alexeygrigorev.com/aihero/

Certificate

Take some time to wrap up your project. Tomorrow we will send you a form, fill it in, and you'll receive a certificate.

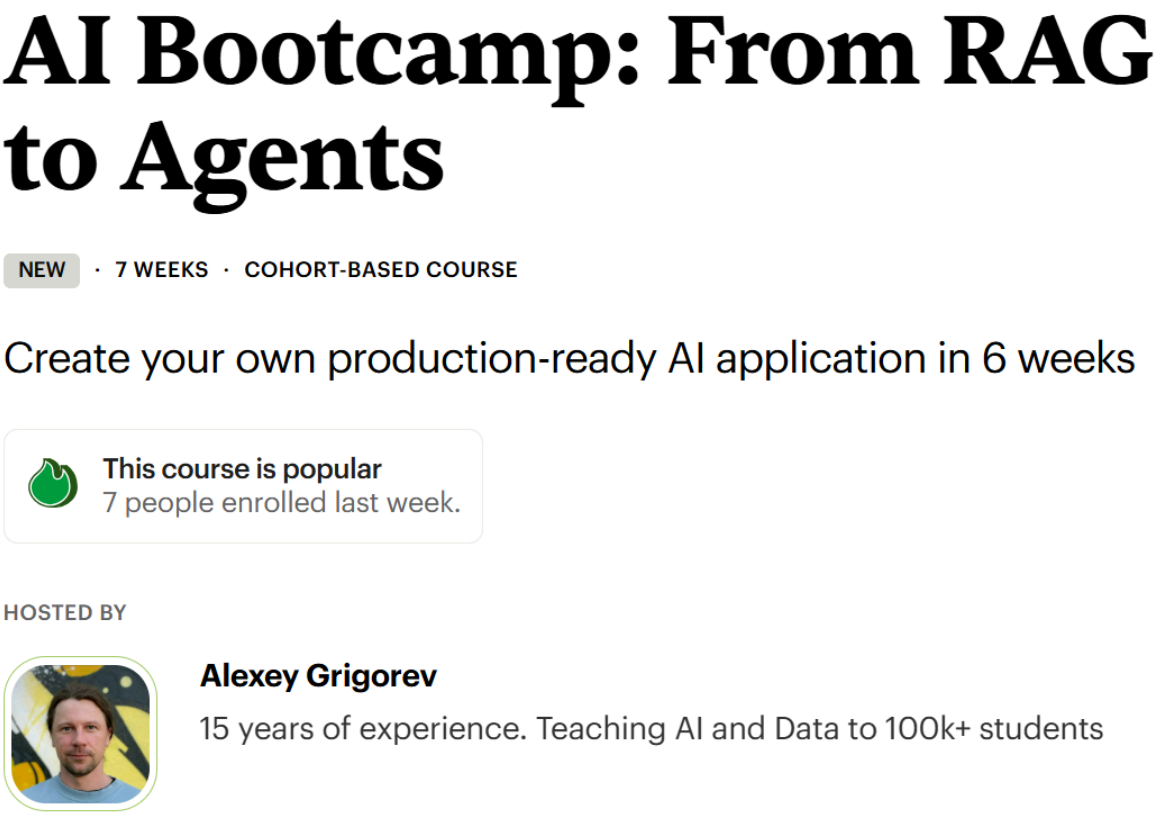
In order to qualify for a certificate, you need to:

- Use a different dataset from the one we used in this course (so it can't be the FAQ dataset)
- Have a data pipeline, agent code, evals and UI
- Organize the code into manageable Python scripts (you can't have everything in one big Jupyter notebook)
- Create a README with a video demo
- Share the end project on your social media, tag me and mention the course.

AI Bootcamp: from RAG to Agents

Congratulations on completing the AI Agents Crash Course! 🎉

If you enjoyed this course, there's more! I'm hosting a bootcamp about building AI Agents and we will do more things there.



In addition to working with GitHub repos, we'll also see how to process YouTube videos, put more emphasis on building data processing pipelines, add tests, cover evaluations in more details, add monitoring and guardrails. You will also have a capstone project and a hackathon at the end!

If you're interested, you can use code "AIHERO" to get 15% off.

Enroll here: https://maven.com/alexey-grigorev/from-rag-to-agents

Community

You can continue the conversation and get support in **DataTalks.Club Slack**:

- Join DataTalks.Club
- Find us in the `#course-ai-bootcamp` channel

In the community channel, you can:

- Share your completed projects and get feedback
- Help other learners who are still working through the course
- Discuss advanced AI agent topics
- Network with other AI practitioners
- Get advice on career opportunities in AI

The learning doesn't stop here - the community is always discussing new techniques, sharing resources, and collaborating on projects.

We'd love to see what you've built. Don't hesitate to share your project on the channel!