

## **CS 203: Software Tools and Techniques for AI Assignment 1**

Chinmay Pendse (23110245)  
Bhamare Dakshata(23210027)

GitHub repository: [Assignment\\_1](#)

### **Introduction:**

In this assignment; we were given a ready-made codebase which we had to edit and make it functional as instructed. We used softwares of VSCode to edit the files and used a Jaeger exporter through the open telemetry library in Python. We had to use Docker to connect our python code to the Jaeger webpage on localhost:16686. In the assignment, we were asked to do these things.

- 1. Add Courses to the Catalog**
- 2. OpenTelemetry Tracing**
- 3. Exporting Telemetry Data to Jaeger**

We have done all the required and the screenshots are being attached in the next section.

### **1 Adding courses to the catalog**

In the attached screenshot, we have implemented the “Add Course” button ; which redirects to the Add Course Page. We have also tabulated the course data which is being derived from couse\_catalog.json.

Course Portal Home Course Catalog

## Course Catalog

Add Course

Course Code	Course Name	Instructor	Semester	Schedule	Classroom	Prerequisites	Description
CS101	<a href="#">Introduction to Computer Science</a>	Dr. Smith	Fall 2024	Mon, Wed, Fri 10:00-11:00 AM	Room 101	None	An introduction to the basics of computer science.
CS 203	<a href="#">Software and Tools for AI</a>	Prof. Mayank Singh	Fall 2025	Mon, Wed, Fri 10:00-11:00 AM	AB 7/109	Basic Python, Linux	
ES 667	<a href="#">Deep Learning</a>	Anirban Dasgupta	Fall 2025	Tue and Thurs: 3:30 - 4:50 PM	AB10/201	Machine learning course at IITGN (ES 335)	An introduction to deep learning techniques.
Es245	<a href="#">control System</a>	rajendran					
es242	<a href="#">dsa</a>	neeldhaara					
ph201	<a href="#">electrodynamics</a>						
cs202	<a href="#">stt cse</a>						

Upon successful submission of the form; the course is being added to the course\_catalog.json and thus it is also visible in the course catalog. The user gets a message regarding the same as a green signal as shown in the next screenshot.

Course Portal Home Course Catalog

Course 'Machine Learning' added successfully!

## Course Catalog

Add Course

Course Code	Course Name	Instructor	Semester	Schedule	Classroom	Prerequisites	Description
CS101	<a href="#">Introduction to Computer Science</a>	Dr. Smith	Fall 2024	Mon, Wed, Fri 10:00-11:00 AM	Room 101	None	An introduction to the basics of computer science.
CS 203	<a href="#">Software and Tools for AI</a>	Prof. Mayank Singh	Fall 2025	Mon, Wed, Fri 10:00-11:00 AM	AB 7/109	Basic Python, Linux	
ES 667	<a href="#">Deep Learning</a>	Anirban	Fall 2025	Tue and	AB10/201	Machine learning	An introduction to

However, if the user does not fulfill the required fields ; he is being given a red signal as shown below; and the requirements are being told to the user. The data regarding incorrect form submission is being recorded to app.json and thus on the jaeger exporter.

The screenshot shows a web browser window with the address bar displaying `http://127.0.0.1:5000/add_course`. The browser's tab bar shows several open tabs, including 'Pen', 'Sch', 'Ger', 'Fre', 'CS2', 'hdf', 'STA', 'STA', 'whi', 'Lab', and 'Cou'. The browser's address bar also shows a star icon, a download icon, and a 'P' icon. The browser's bookmarks bar shows 'All Bookmarks'.

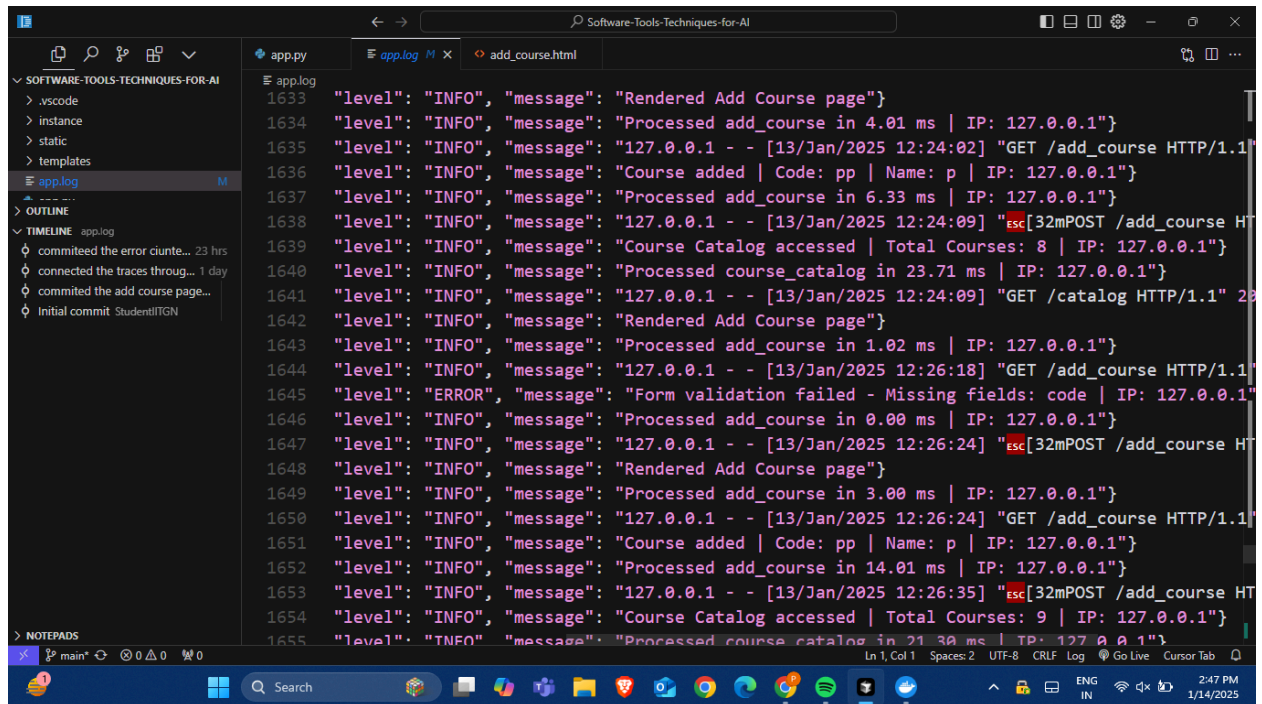
The web page has a dark header with the text 'Course Portal' and links for 'Home' and 'Course Catalog'. Below the header, there is a red error message box that says 'Missing fields: code, name'. Below the error message, the title 'Add a New Course' is displayed. Below the title, there is another red error message box that says 'Missing fields: code, name'. Below the error message, there are six input fields with labels: 'Course Code:', 'Course Name:', 'Instructor:', 'Semester:', 'Schedule:', and 'Classroom:'.

---

## 2. Open Telemetry Tracing

In this assignment we have used Open-Telemetry and logging module such that we can trace what actions are being taken by the user while using the website; We can get the information about the user like the host, the page visited, the request methods etc. on the app.log file which

is being created and updated as soon as someone uses the website. The app.json looks like this.



The screenshot shows a Visual Studio Code editor window with the file 'app.log' open. The log contains a series of JSON objects representing application events. The events include page rendering, course processing, course catalog access, and form validation errors. The log is timestamped with dates from January 13, 2025, and includes IP addresses for each request. The editor interface shows the file explorer on the left with 'app.log' selected, and the timeline view below it showing commit history. The status bar at the bottom indicates the current file is 'main' and the encoding is UTF-8.

```
1633 "level": "INFO", "message": "Rendered Add Course page"}
1634 "level": "INFO", "message": "Processed add_course in 4.01 ms | IP: 127.0.0.1"}
1635 "level": "INFO", "message": "127.0.0.1 - - [13/Jan/2025 12:24:02] \"GET /add_course HTTP/1.1\" 200 127.0.0.1"}
1636 "level": "INFO", "message": "Course added | Code: pp | Name: p | IP: 127.0.0.1"}
1637 "level": "INFO", "message": "Processed add_course in 6.33 ms | IP: 127.0.0.1"}
1638 "level": "INFO", "message": "127.0.0.1 - - [13/Jan/2025 12:24:09] \"esc[32mPOST /add_course HTTP/1.1\" 200 127.0.0.1"}
1639 "level": "INFO", "message": "Course Catalog accessed | Total Courses: 8 | IP: 127.0.0.1"}
1640 "level": "INFO", "message": "Processed course_catalog in 23.71 ms | IP: 127.0.0.1"}
1641 "level": "INFO", "message": "127.0.0.1 - - [13/Jan/2025 12:24:09] \"GET /catalog HTTP/1.1\" 200 127.0.0.1"}
1642 "level": "INFO", "message": "Rendered Add Course page"}
1643 "level": "INFO", "message": "Processed add_course in 1.02 ms | IP: 127.0.0.1"}
1644 "level": "INFO", "message": "127.0.0.1 - - [13/Jan/2025 12:26:18] \"GET /add_course HTTP/1.1\" 200 127.0.0.1"}
1645 "level": "ERROR", "message": "Form validation failed - Missing fields: code | IP: 127.0.0.1"}
1646 "level": "INFO", "message": "Processed add_course in 0.00 ms | IP: 127.0.0.1"}
1647 "level": "INFO", "message": "127.0.0.1 - - [13/Jan/2025 12:26:24] \"esc[32mPOST /add_course HTTP/1.1\" 200 127.0.0.1"}
1648 "level": "INFO", "message": "Rendered Add Course page"}
1649 "level": "INFO", "message": "Processed add_course in 3.00 ms | IP: 127.0.0.1"}
1650 "level": "INFO", "message": "127.0.0.1 - - [13/Jan/2025 12:26:24] \"GET /add_course HTTP/1.1\" 200 127.0.0.1"}
1651 "level": "INFO", "message": "Course added | Code: pp | Name: p | IP: 127.0.0.1"}
1652 "level": "INFO", "message": "Processed add_course in 14.01 ms | IP: 127.0.0.1"}
1653 "level": "INFO", "message": "127.0.0.1 - - [13/Jan/2025 12:26:35] \"esc[32mPOST /add_course HTTP/1.1\" 200 127.0.0.1"}
1654 "level": "INFO", "message": "Course Catalog accessed | Total Courses: 9 | IP: 127.0.0.1"}
1655 "level": "INFO", "message": "Processed course_catalog in 21.30 ms | IP: 127.0.0.1"}

```

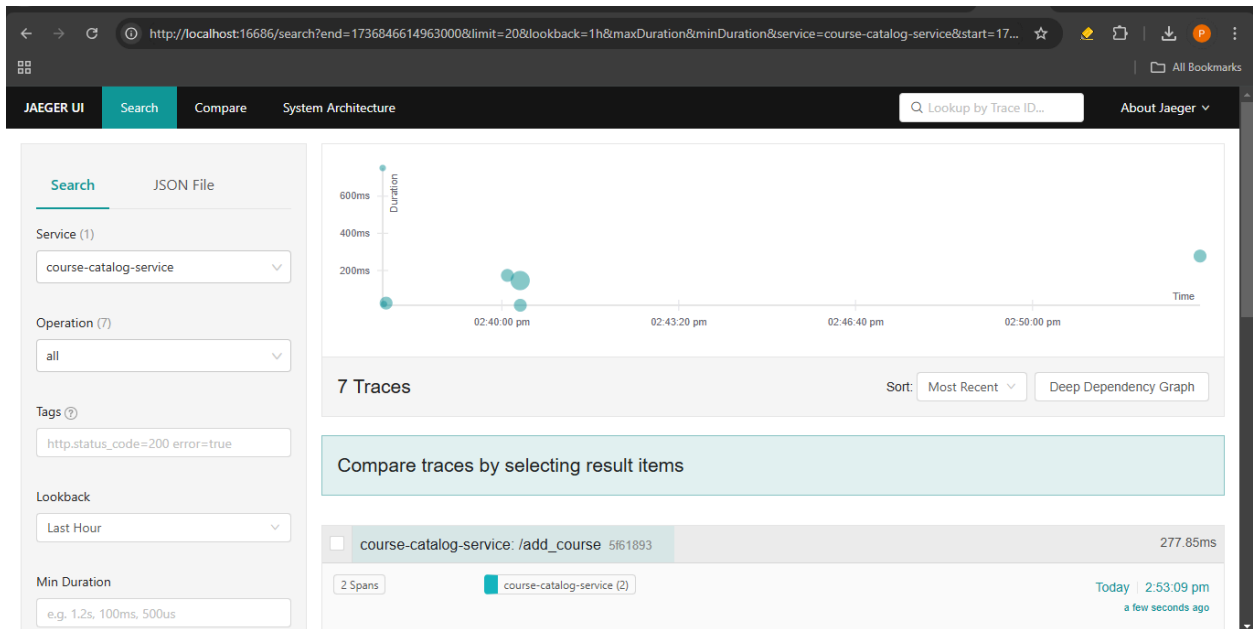
All the browsing is being monitored here. If more courses are added they are being reflected to course\_catlog.json as shown below.

```
app.py  course_catalog.json  add_course.html
course_catalog.json > {} 3
35      {
45      },
46      {
47          "code": "es242",
48          "name": "dsa",
49          "instructor": "neeldhaara",
50          "semester": "",
51          "schedule": "",
52          "classroom": "",
53          "prerequisites": "",
54          "grading": "",
55          "description": ""
56      },
57      {
58          "code": "ph201",
59          "name": "electrodynamics",
60          "instructor": "",
61          "semester": "",
62          "schedule": "",
63          "classroom": "",
64          "prerequisites": "",
65          "grading": "",
```

---

### 3. Exporting Telemetry data to Jaeger.

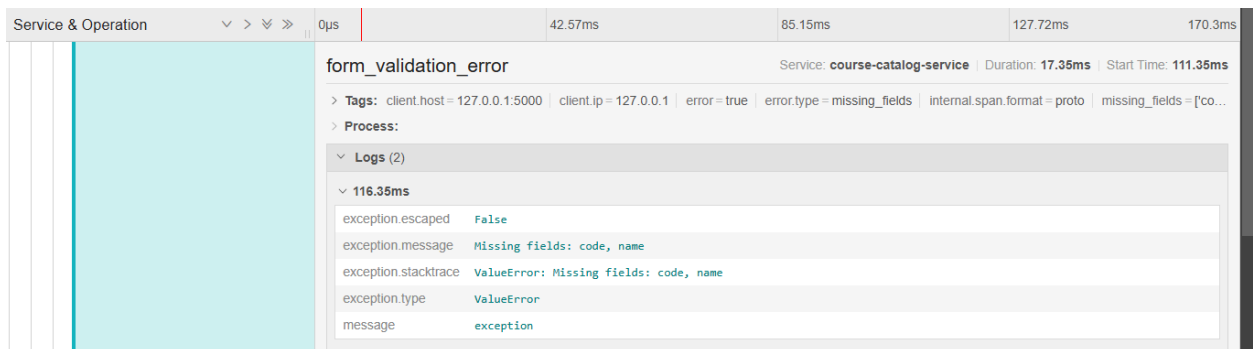
Here, we use the JaegerExporter exporter from telemetry to connect our webpage's logging to Jaeger software. We can access that through localhost:16686. To get all kinds of traces, we set the Operations to all, and we can get the data like this.



Now suppose we want the data of the user adding a wrong course; we get a `form_validation_error` operation and the error is also visible.



If we expand this error we can get the complete data about the user; the number of error counts made; host etc.



The screenshot displays the 'Logs' section of a Jaeger trace. It contains two log entries:

- 116.35ms**:
  - exception.escaped: False
  - exception.message: Missing fields: code, name
  - exception.stacktrace: ValueError: Missing fields: code, name
  - exception.type: ValueError
  - message: exception
- 118.11ms**:
  - client.ip: 127.0.0.1
  - error\_count: 2
  - fields\_missing: ['code', 'name']
  - message: validation\_failed
  - timestamp: 1736846738.4251814

Log timestamps are relative to the start time of the full trace.

SpanID: ee8d6de4a6bdd718

Also, if a new course is added; a new operation is being generated for the meta-data site; we can get the number of traces from there too.

The screenshot shows a trace for the operation `view_course_CS 203` in the `course-catalog-service`. The duration is `607µs` and the start time is `26.96ms`.

**Tags:** client.host = 127.0.0.1:5000 | client.ip = 127.0.0.1 | course.code = CS 203 | course.instructor = Prof. Mayank Singh | course.name = Software a...

**Process:**

- Logs (1)**:
  - 26.96ms**: client.ip = 127.0.0.1 | course.code = CS 203 | message = course\_accessed | timestamp = 1736847010.3270056

Log timestamps are relative to the start time of the full trace.

SpanID: 06b7d9bca692f990

This is similar for all the web pages which is made across the assignment and we get the each and every logging data required.

## Conclusions:

- 1) Add course button created to add courses in catalog.
- 2) OpenTelemetry tracing used to get the logging data.
- 3) Logging data exported on the Jaeger software.

