DS-551 Syllabus

DS-551: Data Engineering at Scale

Spring 2026 Syllabus

Course Description & Pedagogical Approach

Welcome to Data Engineering at Scale. This course immerses you in the practice of building and operating systems that collect, move, transform, store, analyze, and publish data at production scale. We organize the semester around the Epidemic Engine — a hypothetical but realistic application that gathers intelligence about potential health events, aggregates heterogeneous data, publishes it in multiple forms, and supports forecasting. Although fictional, the Epidemic Engine mirrors the problems you will face in industry: imperfect data, changing requirements, trade-offs among latency, throughput, and cost, and the need for observability and governance.

Across the term, we will:

- Collect and integrate data from APIs, scrapers, and streams.
- Design and orchestrate robust pipelines using modern batch and streaming paradigms.
- Work across data models (relational, document, graph, geospatial) with an emphasis on interoperability.
- Operate distributed systems (containers, clusters, topics, jobs) and reason about failure, back-pressure, and recovery.
- Publish and monitor outputs with logs, metrics, and alerts to support decision-making.
- Evaluate trade-offs in consistency, availability, performance, governance, and ethics.

The goal is not to memorize tools but to develop documentation-driven engineering habits that make you adaptable as technologies evolve.

Instructional Format, Course Pedagogy, and Approach to Learning

The course is a fast-paced, project-based exploration of contemporary data engineering. The Epidemic Engine provides a unifying context so that lectures, assignments, and projects compose into a coherent whole. With few exceptions, the components you build will integrate into that system.

- Class meetings. Lectures meet Tuesday/Thursday and discussions meet Wednesday (two sections). Meeting times and rooms are listed in MyBU Student. Most meetings will have a short in-class activity that assumes you completed the assigned reading(s) or GenAI exploration (GAIE) beforehand. Lecture time is then focused on the most important concepts, trade-offs, and current events. Slides and references are posted afterward.
- GenAI Explorations (GAIEs). Short, guided pre-class activities using an AI tool (TerrierGPT or equivalent). GAIEs give you first contact with new concepts before lecture consolidates them. See the Assessment section for grading and completion requirements.
- Homeworks. Three substantial engineering assignments (HW01-HW03) that build toward the project. Most are team-based and rubric-graded, with an individual verification component.

- Project. One cumulative project delivered in three phases adds capabilities to the Epidemic Engine. Teams document design decisions, deployment choices, observability, and operational run-books.
- Discussions. Weekly sections led by the TA emphasize Q&A, debugging, design critiques, and short guided exercises.
- Recordings. Class sessions are recorded on a best-effort basis for enrolled students; recordings may not capture every interaction and are not guaranteed.

Attendance is mandatory, as in-class activities and discussions are central to verifying understanding.

Prerequisites

- DS 310 (or equivalent)
- Strong Python programming skills
- Exposure to SQL and DDL
- A GitHub account

Learning Outcomes

By the end of the course, students will be able to:

- Explain the complexity of modern data ecosystems for enterprise-scale applications.
- Apply core concepts and evaluate tools that address data engineering challenges.
- Distinguish conceptual understanding from implementation details of specific tools.
- Adapt to new technologies by effectively reading and applying official documentation.

Learning Modules (High-Level Overview)

Collection + Integration APIs, scraping, and relational integration that establish how data enters the system and becomes queryable.

Containers + Platform Container fundamentals and OpenShift basics needed to deploy repeatable data jobs.

Distributed Architectures Core distributed systems vocabulary and event-driven patterns used to reason about scale and failure.

Storage Formats + Hardware Cloud-native storage formats and hardware trade-offs that shape performance and cost decisions.

Batch Processing MapReduce and Spark foundations for scalable, repeatable batch workflows.

Pipelines + Streaming Pipeline tooling, orchestration concepts, Kafka, and real-time pipeline patterns.

Advanced Integration ML at scale, governance, observability, security/ethics, scaling, and case-study synthesis.

Books & Tools

- Required text: Martin Kleppmann, Designing Data-Intensive Applications.
- Supplemental readings: Selected articles, docs, and whitepapers (posted in Blackboard Ultra).
- Primary environment: Python; GitHub; containers (Docker/Podman); Kubernetes / OpenShift.
- Tools: DS-551 covers concepts including event streaming, containerized pipeline orchestration,

distributed batch processing, and cloud-native storage. Historically the course has used tools such as Apache Kafka, Docker/Podman, Kubernetes, OpenShift Pipelines (Tekton), and Apache Spark. The instructor selects specific implementations each semester based on what best illustrates the underlying concepts.

- AI tools: GenAI tools are used throughout the course — in projects, homeworks, labs, and pre-class explorations (GAIEs) — unless an activity explicitly restricts their use. Students use TerrierGPT or an equivalent tool. Students may need a paid subscription; budget approximately $60 for the semester (treat this as a course materials cost, similar to a supplemental text).
- Cloud costs: No student-paid cloud compute resources are expected.

Courseware

- Blackboard Ultra is the source of truth for the course: announcements, deadlines, lecture materials, office-hours calendar, and links to submission portals.
- Gradescope (via Blackboard) is used for submissions and feedback; access it by clicking the assignment link in Blackboard (direct login is optional).
- Piazza is for Q&A and logistical updates; post public questions when possible so answers benefit the class.
- GitHub hosts course repositories for assignments and projects.

Assessment & Grading

Grade Distribution

Category	Weight
GenAI Explorations (GAIEs)	5%
In-Class Activities	15%
Homeworks (HW01-HW03)	30%
Project	50%

Project peer evaluation is not a separate grade category. Instead, after each project phase, a peer evaluation multiplier (0.8x, 1.0x, or 1.1x) is applied to each student's score for that phase.

GenAI Explorations (GAIEs) (5%)

GAIEs are short, completion-graded, pre-class AI-guided explorations.

- Graded: done / not done. No partial credit.
- When to do them: Complete each GAIE before the associated lecture. If you miss one, you may submit it late with no grade penalty.
- Gate for the final oral interview: All GAIEs must be completed before the final oral interview.
- Submission: Gradescope — tool used, approximate time, one takeaway or confusion.

In-Class Activities (15%)

Short graded activities completed during lecture or discussion. Most are team-based; some are individual.

- Cannot be made up — they are tied to that specific class session.
- Lowest 15% of activity grades are automatically dropped.

Homeworks (HW01-HW03) (30%)

Three substantial engineering assignments that build the Epidemic Engine component by component.

Grading philosophy — completion floor + quality ceiling: A complete, working submission earns a minimum of 80 points. Rubric items for correctness, clarity, reproducibility, and documentation push the score upward.

Individual video verification (HW02 and HW03): Each student submits a short screen-capture video demonstrating and explaining their understanding of the submitted work. This component counts as part of the homework grade and is an individual verification — no GenAI assistance.

Team assignments: HW02 and HW03 are team-based. HW01 is individual.

Project (50%)

One cumulative project delivered in three phases, adding capabilities to the Epidemic Engine.

Phase	Weight of Project	Overall Weight
Phase 1 — API Collection & Integration	25%	12.5%
Phase 2 — Streaming Pipelines	35%	17.5%
Phase 3 — Deployment at Scale	40%	20.0%

Graded by rubric on clarity, correctness, reproducibility, observability, documentation, and teamwork.

Project grading logic:

- After each phase, the team receives a rubric score.
- Each student's score for that phase is adjusted by that phase's **peer evaluation multiplier** (0.8x, 1.0x, or 1.1x).
- The adjusted phase scores are combined using the phase weights above to compute the student's individual project score.
- After the final oral interview, a separate final interview multiplier (0.8x, 1.0x, or 1.1x) is applied to the student's overall project score.

There are no written final exams in this course, but attendance during the final exam period is required. The final oral interview is required for course completion.

Timing & Late Work

- In-class activities: Due in class. No make-ups; missed activities fall under the 15% auto-drop.
- GAIEs: Complete each GAIE before the associated lecture. If you miss one, submit it late — there is no grade penalty for lateness. All GAIEs must be completed before the final oral interview.
- Homeworks: Follow published due dates. Certain deliverables may allow <=48 h late at -10% — see each spec.
- Project (phases): Follow milestone dates precisely. Certain deliverables may allow <=48 h late at -10% — see each spec.
- Excused absences: Approved excused absences receive an automatic 7-day extension before normal late rules apply.
- Late adds: Students who add the course late have a 10-day grace window from their official add date before normal rules apply.

GenAI Policy

GenAI is a productivity enhancer, but the work is still yours. You must be able to demonstrably prove your understanding of any material where GenAI provided assistance.

Context	Examples (non-exhaustive)	Notes
Encouraged	Brainstorm data models and pipeline sketches; draft manifests; summarize docs; rubber-duck debugging	Cite all meaningful AI assistance in your usage log.
Allowed with care	Generate scaffolding code; produce config/templates; refactor for clarity; propose observability metrics	You are responsible for correctness, licensing, and being able to explain every decision.
Prohibited	Video verification recordings, oral interviews, and any activity explicitly marked no-AI	Violations are academic misconduct.

We follow BU's Academic Conduct Code and the CDS GAIA policy, adapted for this course. Always cite collaborators, data, libraries, and GenAI use.

GenAI Zones

- Green (Encouraged): GAIEs, concept exploration, debugging help, project brainstorming.
- Yellow (Allowed with care): Project execution (analysis must be student-driven), writing drafts, study groups.
- Red (Prohibited): Video verification recordings, oral interviews, and any activity or assessment explicitly marked no-AI or "individual verification."

Academic Integrity & Licensing

We follow BU's Academic Conduct Code. Collaboration is encouraged where specified, but all submissions must cite collaborators, data sources, libraries, and GenAI assistance. Respect open-source licensing — no license no reuse.

How to Succeed

1. Do the GAIE before class. Lecture assumes prior exposure.
2. Treat assignments as project scaffolding. Build reusable components.
3. Contribute in lecture. Substantive questions and comments are valued.
4. Document decisions and trade-offs as you go. Automate where possible.
5. Use office hours and Piazza actively. Ask focused questions and share minimal reproducible examples.
6. Build team habits. Small PRs, code reviews, and clear commit messages.
7. Do not skip the final exam period. The Phase 3 interview is a required individual component.

Support & Accessibility

- Disability accommodations: Office for Disability Services (617-353-3658, access@bu.edu). Share your letter privately with the instructor.
- Student support: ERC tutoring, writing resources, and BU Student Wellbeing services (see

Blackboard for links).

Regrades

Submit regrade requests within 7 days of score release via Gradescope (accessed through Blackboard). Scores may go up, down, or remain unchanged; decisions are final.

Last updated: 2026-03-23