Spring 2025 LeetCode Bootcamp Syllabus

Welcome to the NYU Tandon Spring 2025 LeetCode Bootcamp! Over the next nine weeks, students will gain hands-on experience practicing LeetCode questions in a supportive group environment to learn the fundamentals of answering technical interviewing questions. Here is more information about what to expect during these sessions:

Instructor Information: Kumuda Aggarwal

All sessions are virtual, and students who register will receive calendar invites with unique Zoom links. Students must log in using their NYU credentials for us to keep track of individual participation during the Bootcamp.

Please email leetcodebootcamp@nyu.edu if you have any questions or comments.

Teaching Sections

Two teaching sections are available; the times and dates are listed below. A different topic will be delivered each week. **Participants must stay with their chosen teaching section**.

Section 1: Tuesdays (12:00 PM - 2:00 PM EST)

February 25
March 4, 11, 18
April 1, 8, 15, 22, 29

Section 1 Zoom link

Section 2: Thursdays (12:00 PM - 2:00 PM EST)

February 27
March 6, 13, 20
April 3, 10, 17, 24
May 1

Section 2 Zoom link

Please fill out this teaching section feedback form each week

Bootcamp Topics & Timeline

Week 1 (2/25 & 2/27): Lists, Arrays, Sorting

Week 2 (03/04 & 03/06): Strings, Hash Maps

Week 3 (03/11 & 03/13): Matrix, Linked Lists

Week 4 (03/18 & 03/20): Graph, Stack, Queue

Week 5 (04/01 & 04/03): Mid Program Project Presentations

Week 6 (04/08 & 04/10): Tree, Trie, Heap

Week 7 (04/15 & 04/17): Search Algorithms: Depth-First search, Breadth-First search, Binary Search Tree

Week 8 (04/22 & 04/24): Greedy Algorithms, Dynamic Programming

Week 9 (04/29 & 05/01): Final Project Presentations

Pre-Bootcamp Resources

The NYU Tandon LeetCode Bootcamp is open to all NYU Tandon students, regardless of coding proficiency level. Please consult the following resources to help you better understand technical interviews and practice coding in your free time while participating in the Bootcamp:

Tech Interview Handbook
NeetCode - YouTube
CodeSignal
Codility
Visualizing Algorithms
Tandon Technical Interview Guide

How each session works

1. **Topic introduction:** The instructor will spend 10 to 15 minutes providing background on the week's topic.
2. **Solving Leetcode problems:** Several problems, from easy to challenging, will be discussed. Students will have 15-30 minutes to resolve these problems accordingly.
3. **Facilitate Q&A:** Students can discuss the underlying code and the optimum explanatory process. They may raise questions in the chat or by unmuting themselves.
4. **Take-home assignments:** Please see below for more information regarding take-home exercises.
5. **Breaks:** Students will be given several breaks throughout each session.

Take-home assignments

Students will be given problems related to the week's topic to work on at home each week. Students may work individually or together to solve take-home problems, but must submit problems individually.

Please use this submission form to share your work and results each week by Tuesdays/Thursdays at 11:59PM EST!

Please review the following GitHub link for take-home assignments and other pertinent content throughout the boot camp

Assignments are expected to be submitted by the due date. Students who miss a class are expected to still submit their assignments on time. If you need additional time beyond the submission deadline to complete an assignment, email leetcodebootcamp@nyu.edu.

Industry Sessions

As an added bonus to topics covered in teaching sessions, the bootcamp will feature two presentations from industry practitioners about presenting work as a software engineer and the software engineering hiring process.

Presenting Work as a Software Engineer 03/17, 1-2PM, Virtual
In this presentation, students will hear from a software engineering industry practitioner about tips and tricks for navigating professional software engineering projects, communicating technical concepts to non-technical audiences, and how to present their work effectively.

The Software Engineering Hiring Process, 04/07, 1-2PM, Virtual
In this presentation, students will hear from a software engineering professional about the ins and outs of the recruiting process for software engineering roles, what companies look for in potential software engineers, and how navigating technical interviews help land their next job or internship.

Project

Students selected for the Spring 2025 LeetCode Bootcamp will work on a real-world project to apply the skills and knowledge gained during the bootcamp.

Past projects have included utilizing skills like building web applications, designing user interfaces and other relevant tools and techniques!

Projects are **required** as part of student participation in the LeetCode Bootcamp. Students may work with each other in groups of **5-6 students** to complete the project and create their groups by **Friday, March 7 @ 11:59PM**.

Project groups will conduct their **mid-program presentations** with industry professionals for 10 minutes (presentation and project judge Q&A) on **April 1** and **April 3** (whichever date corresponds with your teaching section) to receive feedback and make revisions for the final project presentations.

Groups will conduct their **final project presentations** with industry professionals for 10 minutes (presentation and project judge Q&A) the week of **April 29 and May 1** (whichever date corresponds with your teaching section). Prizes will be awarded to the winning group.

Students are encouraged to participate in Slack discussions and to communicate with the instructor regularly about their project progress, share ideas, or ask questions.

Project Judging Criteria

Projects will be judged on the following criteria, and prizes will be awarded to the top-performing projects:

Using 1 (lowest) - 3 (highest)

Project Design: How was the project designed, and did students ask questions that helped move the project forward proactively?

- 1: Project not designed well; yields no tangible results or applicability
- 2: Project designed to move forward, but disjointed approach
- 3: Project designed to move forward systematically

Addressing Stated Problem: How well does the project solve the stated problem?

- 1: Results did not address the stated problem
- 2: Results include missing details and inconclusive results
- 3: Results solve the problem at face value

Presentation and Demo: How was the presentation?

- 1: The project presentation lacked any sort of depth and clarity; the student(s) presenting did not address issues from prompt
- 2: The project presentation included little detail and inconclusive results, difficult to understand
- 3: The project presentation addresses the challenges stated in the prompt clearly and concisely, making it easy for the audience to follow and understand

Project Options

Below you will find the project options for the Spring 2025 LeetCode Bootcamp. Each group must choose one project by **Friday, March 7 @ 11:59PM** date. For each option, we offer suggested key features and a suggested timeline. These are merely suggestions to offer guidance for the project, you may pursue different features and a timeline that works best for your group.

Project Name: Scholarly Insight

Objective: To develop a web application that allows users to search, browse, and stay updated with scholarly articles across various scientific disciplines provided by the arXiv API.

API: https://info.arxiv.org/help/api/index.html

Suggested Key Features:

- Search and Browse: Functionalities to search and browse articles using author, category, and publication date filters.
- Article Details: Display detailed article information, including abstracts and links to full papers.
- User Accounts: Allow users to create accounts to save favorite articles, set up alerts for new publications, and set up user authentication using Firebase.
- Reading History: Track and display users' reading histories.
- Notifications: Implement a system for email or in-app notifications for new articles in user-specified categories.
- Discussion Forum: A space for users to discuss articles and share insights.

Suggested Timeline:

- ➔ Week 1: Setup
  - ◆ Task 0: Team formation (4-5 students)
  - ◆ Task 1: Explore arXiv API
- ➔ Week 2: Initial Development
  - ◆ Task 2: Set Up Development Environment
  - ◆ Task 3: Design Database Schema and Set Up Backend Framework
- ➔ Week 3-4: Backend Development
  - ◆ Task 4: Implement Article Search and Browse Backend
- ➔ Week 5: Mid-Checkpoint Presentation
  - ◆ Present backend development progress, API data fetching strategy, and search functionality.
- ➔ Week 6: Frontend Development and Integration
  - ◆ Task 5: Develop Frontend Structure and Implement Article Details
  - ◆ Task 6: Integrate Firebase and Start Discussion Forum Implementation
- ➔ Week 7: Finalizing Features and Testing
  - ◆ Task 7: Complete the Discussion Forum and Implement Notifications
  - ◆ Task 8: Testing and Debugging
- ➔ Week 8: Deployment and Final Presentation Preparation
  - ◆ Task 9: Deployment
  - ◆ Task 10: Final Testing and Presentation Preparation
- ➔ Week 9: Final Presentation

Project Name: DineWise

Objective: Develop a web application that allows users to search, compare, and review local restaurants and eateries using data from the Yelp API

API: Yelp Fusion API

Suggested Key Features:

- Search and Browse: Users can search for restaurants with filters based on location, cuisine type, ratings, and price range.
- Restaurant Details: Display detailed information for each restaurant, including reviews, ratings, photos, and links to the Yelp page.
- User Accounts: Registered users can rate and review restaurants. Implement user authentication using Firebase
- Store Locator: Feature to locate nearby restaurants
- Wishlist: Allow users to create and manage a wishlist of their favorite restaurants.
- Event and Promotion Notifications**:** Notify users of special events, promotions, or discounts at their favorite restaurants based on their wishlist or past visits.

Suggested Timeline:

- ➔ Week 1: Setup
  - ◆ Task 0: Team formation (4-6 students)
  - ◆ Task 1: Explore Yelp API
- ➔ Week 2: Initial Development
  - ◆ Task 2: Set Up Development Environment
  - ◆ Task 3: Design Database Schema and Set Up Backend Framework
- ➔ Week 3-4: Backend Development
  - ◆ Task 4: Implement Restaurant Search and Filters Backend
- ➔ Week 5: Mid-Checkpoint Presentation
  - ◆ Present backend development progress, including restaurant comparison and store locator
- ➔ Week 6: Frontend Development and Integration
  - ◆ Task 5: Develop Frontend Structure and Implement Restaurant Search
  - ◆ Task 6: Integrate Firebase and Implement Restaurant Comparison
- ➔ Week 7: Finalizing Features and Testing
  - ◆ Task 7: Implement Reservation System and Store Locator
  - ◆ Task 8: Testing and Debugging
- ➔ Week 8: Deployment and Final Presentation Preparation
  - ◆ Task 9: Deployment
  - ◆ Task 10: Final Testing and Presentation Preparation
- ➔ Week 9: Final Presentation

Project Name: Movie Magic

Objective: To develop a web application that allows users to discover, browse, and review movies and TV shows, utilizing data from The Movie Database (TMDb) API.

API: TMDb API

Suggested Key Features:

- Movie and TV Show Search: Enable users to search for movies and TV shows by title, genre, or release date.
- Detailed Information: Display detailed information for each movie or TV show, including summaries, ratings, cast, crew, and trailers.
- User Reviews**:** Allow users to write and submit their own reviews for movies and TV shows.
- User Accounts: Enable users to create accounts to save favorite articles, set alert preferences, and manage their profiles. Use Firebase for authentication.
- Trending Content: Provide sections for trending movies, popular shows, and upcoming releases.
- Favorites List**:** Users can create and manage a favorites list to save movies and shows they want to revisit.

Suggested Timeline:

- ➔ Week 1: Setup
  - ◆ Task 0: Team formation (4-6 students)
  - ◆ Task 1: Explore TMDb API and understand the endpoints.
- ➔ Week 2: Initial Development
  - ◆ Task 2: Set Up Development Environment
  - ◆ Task 3: Design Database Schema and Set Up Backend Framework
- ➔ Week 3-4: Backend Development
  - ◆ Task 4: Implement Movies Search and Filters Backend
- ➔ Week 5: Mid-Checkpoint Presentation
  - ◆ Present backend development progress, API data fetching strategy, and search functionality.
- ➔ Week 6: Frontend Development and Integration
  - ◆ Task 5: Develop Frontend Structure and Implement Restaurant Search
  - ◆ Task 6: Integrate Firebase and Implement review feature
- ➔ Week 7: Finalizing Features and Testing
  - ◆ Task 7: Complete the user reviews feature and implement the favorites list.
  - ◆ Task 8: Testing and Debugging
- ➔ Week 8: Deployment and Final Presentation Preparation
  - ◆ Task 9: Deployment
  - ◆ Task 10: Final Testing and Presentation Preparation
- ➔ Week 9: Final Presentation

Project Name: Congress Tracker

Objective: Develop a web application to track and display U.S. legislative data using the Congress API.

API: https://github.com/LibraryOfCongress/api.congress.gov

Suggested Key Features:

- Legislative Dashboard: View recent bills and implement advanced search options with filters like date range, bill status, and keyword.
- Member Profiles: Profiles for Congress members with their bill sponsorship and voting records.
- User Accounts: Basic account creation and user authentication using Firebase.
- Notifications: Alerts for updates on tracked bills or members.
- Data Visualization: Simple graphs for legislative data.
- Community Engagement: Implement a feature for users to comment on and discuss bills.

Suggested Timeline:

- ➔ Week 1: Setup
  - ◆ Task 0: Team formation (4-6 students)
  - ◆ Task 1: Explore Congress API
- ➔ Week 2: Initial Development
  - ◆ Task 2: Set Up Development Environment
  - ◆ Task 3: Design Database Schema and Set Up Backend Framework
- ➔ Week 3-4: Backend Development
  - ◆ Task 4: Implement Legislative Dashboard and Member Profiles Backend
- ➔ Week 5: Mid-Checkpoint Presentation
  - ◆ Present the backend development progress, database schema, and initial API integration.
- ➔ Week 6: Frontend Development and Integration
  - ◆ Task 5: Develop a Frontend Structure and Implement a Legislative Dashboard
  - ◆ Task 6: Implement Member Profiles and Integrate Firebase Authentication
- ➔ Week 7: Finalizing Features and Testing
  - ◆ Task 7: Implement Notifications and Data Visualization
  - ◆ Task 8: Testing and Debugging
- ➔ Week 8: Deployment and Final Presentation Preparation
  - ◆ Task 9: Deployment
  - ◆ Task 10: Final Testing and Presentation Preparation
- ➔ Week 9: Final Presentation

Project Name: NYC Transit Hub

Objective: Develop a web application to offer real-time updates, schedules, and transit information for New York City's public transportation system

API: https://api.mta.info

Suggested Key Features:

- Interactive Map: Display a real-time map of subway and bus routes, allowing users to visualize transit options
- Favorites and Alerts: Allow users to save favorite routes/stations and set up notifications for service changes or delays
- Accessibility Information: Include details about elevator and escalator availability at stations for accessibility.
- Service Status Dashboard: Show an overview of the status of all MTA services in a single dashboard.
- Multilingual Support: Offer the application in multiple languages.
- User Accounts: Basic account creation and user authentication using Firebase.

Suggested Timeline:

- ➔ Week 1: Setup
  - ◆ Task 0: Team formation (4-6 students)
  - ◆ Task 1: Explore MTA API
- ➔ Week 2: Initial Development
  - ◆ Task 2: Set Up Development Environment
  - ◆ Task 3: Design Database Schema and Set Up Backend Framework
- ➔ Week 3-4: Backend Development
  - ◆ Task 4: Implement Live Transit Updates Backend
- ➔ Week 5: Mid-Checkpoint Presentation
  - ◆ Present backend development progress, including real-time data handling and route planning functionality.
- ➔ Week 6: Frontend Development and Integration
  - ◆ Task 5: Develop Frontend Structure and Implement status dashboard
  - ◆ Task 6: Integrate Firebase and Implement transit options
- ➔ Week 7: Finalizing Features and Testing
  - ◆ Task 7: Implement multilingual language support
  - ◆ Task 8: Testing and Debugging
- ➔ Week 8: Deployment and Final Presentation Preparation
  - ◆ Task 9: Deployment
  - ◆ Task 10: Final Testing and Presentation Preparation
- ➔ Week 9: Final Presentation

Suggested Technology Stack

- **Frontend:**
  - Framework: Use any free and open-source JavaScript framework.
  - Styling: Utilize CSS framework for rapid UI development. Ensure the application is mobile-friendly. (Guide)
- **Backend:**

    - Language: Python with Flask (lightweight and easy-to-use web framework).
    - Database: SQLite (lightweight, file-based database, suitable for small projects).
- **API:** Ensure compliance with the chosen API's usage limits.
- **Testing:** Introduce basic unit and integration testing using tools like Jest (for JavaScript) or PyTest (for Python). (Guide)
- **Deployment:** Netlify or Vercel (free hosting options).
- **Version Control:** Git (for version control) and GitHub (for repository hosting).
- **Security:** Implement basic security measures for user data.
- **Documentation:** Keep code and API usage documented for maintainability. (Guide)

Digital Credential/Program Completion Requirements

Students will be issued a verifiable digital credential badge via Accredible (shareable on LinkedIn and other platforms) provided that they meet the following participation criteria:

- Attend at least 7 live sessions; attendance will be taken each session.
    - Attendance will be monitored based on the duration logged on during each teaching section.
    - If a student must arrive late to a session or leave early, they must receive prior approval by emailing us at leetcodebootcamp@nyu.edu.
    - Leaving a session early or coming late without prior approval will result in the student being marked absent.
- Submit weekly **take-home problem** solutions via submission form.
- Submit weekly **teaching section feedback form**.
- Students must **form project groups** (5-6 members) and communicate the group members' names and the prompt chosen to the instructor by **Friday, March 7 @ 11:59PM**
- Groups must conduct their **mid-program project presentation** for 10 minutes on **April 1** or **April 3** (whichever date corresponds with your teaching section) to receive feedback and make revisions for the final project presentations.
- Groups must conduct their **final project presentation** on **April 29** or **May 1** (whichever date corresponds with your teaching section).
- It is required that all group participants be present during the presentation; it is up to the group to determine who the presenters are.

Slack Channel

The NYU Tandon LeetCode Bootcamp has a dedicated Slack channel. Students are highly encouraged to participate in discussions with the Bootcamp instructor and each other to have questions related to weekly topics, take-home assignments, and projects answered.

The instructor will monitor the Slack channel during the following periods:

Tuesday: 2pm - 5pm
Thursday: 2pm - 5pm
Friday: 1pm - 3pm

Slack Channel Link:
https://join.slack.com/t/spring2025lee-nrg5055/shared_invite/zt-2y2lvx3s1-Bnkf7BxyzBi0mO8TGWX1vg

One on One Technical Interviewing Coaching

Instructors will be available for scheduled 30 minute coaching sessions to assist students with technical interviewing questions and practice. Please see the following Calendly link for availability:

Calendly link:
https://calendly.com/ka3535-nyu/leetcode-technical-interview-coaching-mock-interview

Students must be logged in using their NYU credentials in order for us to keep track of individual participation during the course of the Bootcamp.