

Spring 2024 LeetCode Bootcamp Syllabus

Welcome to the NYU Tandon Spring 2024 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:

Teaching Sections	3
Section 1: Tuesdays (2:00 PM - 4:00 PM EST)	3
Section 2: Wednesdays (12:00 PM - 2:00 PM EST)	3
Timeline	3
Pre-Bootcamp resources	4
How each session works	4
Review Time	4
Industry Sessions	5
Take-home assignments	5
Project	6
Project Judging Criteria	7
Project descriptions	8
Project Name: Congress Tracker	8
Project Name: Scholarly Insight	9
Project Name: Weather Wiz	10
Project Name: NYCTransit Hub	11
Project Name: TechCompare	12
Suggested Technology Stack	13
Certificate of Completion Requirements	14
Slack Channel	15

Instructor Information: Spriha Jha

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 boot camp.

Please email leetcodebootcamp@nyu.edu if you have any questions or comments, and we look forward to your participation!

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 (2:00 PM - 4:00 PM EST)

February 27

March 5, 12, 26

April 2, 9, 16, 23, 30

[Section 1 Zoom Link](#)

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

February 28

March 6, 13, 27

April 3, 10, 17, 24

May 1

[Section 2 Zoom Link](#)

Please fill out this [teaching section feedback form](#) each week

Timeline

Week	Dates	Topic
Week 1	2/27 2/28	Lists, Arrays, Sorting
Week 2	3/5 3/6	Strings, Hash Maps
Week 3	3/12 3/13	Matrix, Linked Lists
Week 4	3/26 3/27	Graph, Stack, Queue
Week 5	4/2 4/3	Mid-Program Project Presentations
Week 6	4/9 4/10	Tree, Trie, Heap
Week 7	4/16 4/17	Search Algorithms: DFS, BFS, BST
Week 8	4/23 4/24	Greedy Algorithms Dynamic Programming
Week 9	4/30 5/1	Final Project Presentations

Pre-Bootcamp resources

The NYU Tandon LeetCode boot camp 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](#)

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.

Review Time

Students will have the opportunity throughout the Spring 2024 LeetCode Bootcamp to schedule 10-minute 1:1 meetings with the boot camp instructor to ask questions and get feedback on topics covered during the bootcamp, questions about review problems, and issues/questions/progress updates about their group project. Students are encouraged to use these sessions to enhance their learning related to the boot camp. The instructor will be available for these individual meetings during the following times throughout the boot camp:

Please use this [link](#) to schedule 10-minute 1:1 meetings with your instructor outside of teaching sections each week at your convenience.

Industry Sessions

As a bonus to topics covered in teaching sessions, the Spring 2024 LeetCode Bootcamp will feature two presentations from industry practitioners about presenting work as a software engineer and acing software engineering technical interviews. We will also have a panel discussion with recruiters about the hiring process for software engineers.

Dates:

Event Type	Event Name	Date
Presentation 1	<p>Presenting Work as a Software Engineer:</p> <p>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.</p>	April 12, 1:00 PM-2:00 PM (Virtual)
Presentation 2	<p>Acing Software Engineering Interviews:</p> <p>In this presentation, students will hear from a software engineering professional about navigating technical interviews related to software engineering, best practices for articulating their process, and how to showcase their results to help land their next great job or internship.</p>	April 19, 1:00 PM-2:00 PM (Virtual)
Recruiter Panel Discussion	<p>The Software Engineering Hiring Process:</p> <p>In this interactive panel discussion, students will have the opportunity to hear from technical recruiters and hiring professionals about the ins and outs of the recruiting process for software engineering roles, what companies look for in potential software engineers, and best practices to help them move forward in that process.</p>	April 26, 1:00 PM-2:00 PM (Virtual)

Take-home assignments

Students will be given 3 to 4 problems related to the week's topic to work on at home each week. Students may work individually to solve take-home problems.

Please use this [submission form](#) to share your work and results each week by Mondays at 11:59 PM!

Please review the following [GitHub link](#) for take-home assignments and other pertinent content throughout the boot camp.

Project

Students selected for the Spring 2024 LeetCode Bootcamp can work on a real-world project to apply the skills and knowledge gained during the bootcamp.

Projects are required as part of student participation in the LeetCode Bootcamp. Students may work with each other in groups of **4-5 students** to complete the project and create their groups by **Friday, March 8 @ 11:59 PM**.

Project groups will conduct their **mid-program presentations** with industry professionals for 10 minutes (presentation and project judge Q&A) on either **April 2 or 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 30 or May 1** (whichever date corresponds with your teaching section). Prizes will be awarded to the winning group.

Students are encouraged to utilize weekly individual review time and participate in Slack discussions to communicate with the instructor regularly about their project progress, ideas, or 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 descriptions

(Please select one to work on)

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 (4-5 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: 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: Weather Wiz

Objective: To develop a web application that provides users with detailed weather forecasts, including current conditions, hourly forecasts, and long-term predictions.

API: <https://pirateweather.net/en/latest/API/>

Suggested Key Features:

- Weather Dashboard: Display current weather conditions and forecasts.
- Mapping Library: Integrate [Leaflet.js](#) or [Google Maps](#) for interactive maps.
- Localization: Localize the app based on the user's location.
- Search Functionality: Allow users to search for weather by city, zip code, or coordinates.
- User Customization: Users can create accounts to save favorite locations and set preferences for weather alerts. Set up user authentication using [Firebase](#).
- Interactive Maps: Implement maps for visual weather representations.
- Alerts and Notifications: Push notifications for weather changes or severe weather alerts.

Suggested Timeline:

- Week 1: Setup
 - ◆ Task 0: Team formation (4-5 students)
 - ◆ Task 1: Explore Pirate Weather API
- Week 2: Initial Development
 - ◆ Task 2: Set Up Development Environment
 - ◆ Task 3: Set Up the Backend Framework and Design Database Schema
- Week 3-4: Backend Development
 - ◆ Task 4: Implement Weather Dashboard Backend
- Week 5: Mid-Checkpoint Presentation
 - ◆ Present the backend development progress, including weather data fetching and storage strategy.
- Week 6: Frontend Development and Integration
 - ◆ Task 5: Develop Frontend Structure and Implement Weather Dashboard
 - ◆ Task 6: Integrate Firebase and Implement Interactive Maps
- Week 7: Finalizing Features and Testing
 - ◆ Task 7: Implement Alerts and 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: NYCTransit Hub

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

API: <https://api.mta.info>

Suggested Key Features:

- **Live Transit Updates:** Display real-time updates on subway, bus, and rail services. Implement WebSocket for live transit updates.
- **Route Planner:** Allow users to plan their routes with current transit options.
- **Service Alerts:** Implement notifications for service interruptions or delays.
- **User Accounts:** Enable users to save frequent routes and set alert preferences. Set up user authentication using [Firebase](#).
- **Interactive Transit Maps:** Provide interactive maps showing routes, stops, and service status.
- **Favorite Routes:** Allow users to save frequently used routes.
- **Accessibility Information:** Include details on transit accessibility.
- **Multilingual Support:** Offer the application in multiple languages.

Suggested Timeline:

- Week 1: Setup
 - ◆ Task 0: Team formation (4-5 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 Live Transit Updates
 - ◆ Task 6: Integrate Firebase and Implement Route Planner
- Week 7: Finalizing Features and Testing
 - ◆ Task 7: Implement Service Alerts and Interactive Transit Maps
 - ◆ 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: TechCompare

Objective: Develop a web application that allows users to search, compare, and review tech products using data from the Best Buy API.

API: <https://developer.bestbuy.com/> | <https://fakestoreapi.com/>

Suggested Key Features:

- Product Search and Filters: Functionalities for users to search products, with filters based on categories, brands, prices, etc.
- Product Comparison: Feature to compare multiple products based on specifications and prices.
- User Reviews and Ratings: Registered users can rate and review products. Set up user authentication using [Firebase](#).
- Price Tracking: Functionality to track and alert users for price changes on desired products.
- Store Locator: Feature to find Best Buy stores and check product availability.
- E-commerce Integration: Integrate [Stripe API](#) for a simulated checkout experience.
- Wishlist: Allow users to create and manage a wishlist of products.

Suggested Timeline:

- Week 1: Setup
 - ◆ Task 0: Team formation (4-5 students)
 - ◆ Task 1: Explore Best Buy 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 Product Search and Filters Backend
- Week 5: Mid-Checkpoint Presentation
 - ◆ Present the backend development progress, including product comparison and user review implementation strategy.
- Week 6: Frontend Development and Integration
 - ◆ Task 5: Develop Frontend Structure and Implement Product Search
 - ◆ Task 6: Integrate Firebase and Implement Product Comparison
- Week 7: Finalizing Features and Testing
 - ◆ Task 7: Implement Price Tracking 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

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](#))

Certificate of Completion Requirements

Students will receive a certificate of completion if they meet the following participation criteria:

- Attend at least **six live teaching sections**; attendance will be taken each session.
 - Students must log in within 20 minutes of the start time of the teaching section and stay for the duration of the session.
 - If a student must arrive more than 20 minutes 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.
- Students will be asked to **submit weekly feedback on sessions** to help us improve the program's short- and long-term delivery.
- Students must submit weekly **take-home problem** solutions via the submission form.
- Students must form project groups (4-5 members) and communicate the group members' names and the prompt chosen to the instructor by **Friday, March 8 @ 11:59 PM**.
- Groups must conduct their **mid-program project presentation** for 10 minutes on either April 2 or April 3 (whichever date corresponds with your teaching section).
- Groups must conduct their **final project presentation** on April 30 or May 1 (whichever date corresponds with your teaching section).
- All group members for project presentations (mid-program and final) must speak and present during the project demonstrations.

Slack Channel

The NYU Tandon LeetCode Bootcamp has a dedicated Slack channel. Students are highly encouraged to participate in discussions with the boot camp 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:

- **Tuesdays:** 12:00 PM-2:00 PM
- **Wednesdays:** 10:00 AM-12:00 PM; 2:00 PM-3:00 PM
- **Thursdays:** 9:00 AM-5:00 PM
- **Fridays:** 11:00 AM-1:00 PM

[Slack Channel Link](#)