

Software engineer (fresher)

Company: [Company name]

Location: [Location] | Work mode: [Work mode]

Employment type: Full-time | CTC: [CTC for the full-time role]

Joining: [Joining date] | Apply by: [Apply by]

About [Company name]

[About your company]

About the role

[Company name] is hiring a fresher software engineer to join the [Team] team. You will work alongside experienced engineers on a live product, starting with bug fixes and small tickets and moving to features as you learn the code base. A mentor will review your code and give regular feedback. This role suits a new graduate who likes solving problems with code and wants to learn how production software is built, tested and released.

What you will do

- Pick up small, clearly defined tickets from the sprint board, such as bug fixes and minor changes, and deliver them with guidance from a senior engineer
- Read existing code to find where a change belongs and what else it could affect before you write anything
- Reproduce reported bugs, use logs and a debugger to find the cause, and fix it with a short note on what went wrong
- Write clean, readable code in the team's main language and follow its coding standards and naming conventions
- Write unit tests for the code you change and run the full test suite before you raise a pull request
- Use Git every day: create branches, make focused commits, raise pull requests and resolve small merge conflicts
- Respond to code review feedback, ask about anything unclear and apply what you learn to later changes
- Write simple SQL queries to check or correct data in the application database, with a senior's review before anything touches live data
- Update README files, API notes and short design notes so the next engineer can follow what you built
- Take part in daily stand-ups, sprint planning and retrospectives, and say early when you are blocked

What we are looking for

- Working knowledge of at least one programming language, such as Java, Python, C++, C# or JavaScript, shown through code you have written and can explain
- Understanding of object-oriented programming and common data structures (arrays, lists, maps, stacks and queues) and when to use each
- Basic SQL: SELECT queries with filters, joins and grouping
- Familiarity with Git: clone, branch, commit, push and pull request
- A step-by-step approach to debugging, and the ability to describe what you tried and what you found
- At least one academic, internship or personal project that you built yourself and can walk through in detail

- Clear written and spoken communication, including asking for help early and writing short status updates

Good to have

- Basic knowledge of REST APIs and JSON
- Experience writing unit tests with JUnit, pytest, Jest or a similar framework
- Awareness of Agile and Scrum practices
- Comfort with the Linux command line
- Coding contests, hackathons or open-source contributions
- A public GitHub profile with readable code

Who can apply

Final-year students and graduates in computer science, IT, electronics and related engineering streams (B.E./B.Tech) suit most teams, as do BCA, B.Sc (computer science or IT), MCA and M.Sc candidates. Candidates from other streams are worth considering if they can show working code and clear fundamentals.

How to apply

Apply through this job post by [\[Apply by\]](#). Shortlisted candidates will hear from us about the next steps.

Software engineer intern

Company: [Company name]

Location: [Location] | Work mode: [Work mode]

Employment type: Internship | Duration: [Internship duration] | Stipend: [Stipend for the internship]

Joining: [Joining date] | Apply by: [Apply by]

About [Company name]

[About your company]

About the role

[Company name] is offering a software engineering internship for students who want to work on a real code base. You will take small tasks from a mentor, write and test code, and learn how a team plans, reviews and ships software. Interns who do well may be considered for a pre-placement offer (PPO) or a full-time role.

What you will do

- Fix small, clearly scoped bugs chosen by your mentor
- Write unit tests for existing functions and report the gaps you find
- Build one small feature or internal tool, such as a script, report or admin screen, from requirement to review
- Read one module of the code base and explain to the team how it works
- Follow the setup notes on a fresh machine and correct them where they are wrong
- Show your work to the team in a short demo at the end of the internship

What we are looking for

- Student in the pre-final or final year of a computer science, IT or related degree
- Able to write programs in at least one language and explain how your code works
- Knows arrays, lists, maps and basic object-oriented ideas
- Has used Git, even for a college project
- Can commit to the full internship period, with exam dates shared in advance

Who can apply

Final-year students and graduates in computer science, IT, electronics and related engineering streams (B.E./B.Tech) suit most teams, as do BCA, B.Sc (computer science or IT), MCA and M.Sc candidates. Candidates from other streams are worth considering if they can show working code and clear fundamentals.

How to apply

Apply through this job post by [Apply by]. Shortlisted candidates will hear from us about the next steps.

Screening questions to add to your job post

1. **Name one project you built. Which parts did you write yourself, and which came from a tutorial or teammates?** Separates real ownership from copied or shared work. A specific answer gives you something to probe in the interview.
2. **A program runs without errors but prints the wrong result. What do you do first?** A good answer reproduces the problem with a small input and checks values step by step. Guessing and rewriting are weaker signs.
3. **Which data structure would you use to check quickly whether a name already exists in a large set of names, and why?** Tests whether the candidate links a data structure to a real need instead of reciting definitions. Look for a set or hash map and a reason.
4. **Given a table `orders(order_id, customer_id)`, write a SQL query that returns each `customer_id` with more than one order.** Look for grouping by `customer_id` and filtering with `HAVING COUNT(*) > 1`. Accept any other correct query.
5. **Two developers change the same lines of a file. What happens in Git, and how do you fix it?** A good answer mentions a merge conflict and choosing or combining both changes before committing. Candidates who have only worked alone may not know.
6. **Tell us about a time a teacher, teammate or senior pointed out a mistake in your code. What did you change?** Shows how the candidate takes feedback, which matters more in a first job than any single tool.