How to Get an Internship in 2nd Year PDF

To get an internship in 2nd year, build one usable skill (web, data, testing or design), finish one or two small projects with live links, make a one-page resume, and apply widely: startups, faculty research, the AICTE internship portal, LinkedIn and alumni referrals. Start applying two to three months before you want to begin, and never pay a fee for an internship.

Key takeaways

- Second-year interns are hired mostly for one practical skill plus proof of work, not for CGPA alone.
- Startups, faculty research groups and NGOs are more open to 2nd-year students than large companies.
- A short, personal cold email with a project link often works better than mass applications.
- Keep a tracker of every application and follow up once after a week.
- A genuine internship never asks you to pay to join.

Is a 2nd-year internship realistic?

Yes, but the kind of internship is different. Large companies usually run formal internship programmes for 3rd-year students, often through campus. In 2nd year, most opportunities come from startups, small companies, faculty research projects, college incubators, NGOs and open source communities. These roles may be short, remote or unpaid, and that is fine. What you want from a 2nd-year internship is real work to talk about, a reference and a better project on your resume.

If you do not get one, that is normal too. A strong personal project over the summer is almost as useful at this stage.

Step 1: pick one skill you can show

Recruiters for 2nd-year interns ask one question: "Can this student do a small task for us?" So choose one skill and get to a basic working level:

- **Web development:** HTML, CSS, JavaScript and one framework or backend.
- **Data analysis:** Excel, SQL, Python with pandas and simple charts.
- **Software testing:** writing test cases, bug reports and basic automation.
- **UI/UX design:** Figma, wireframes and one case study.

- **Content or technical writing:** clear blog posts or documentation on technical topics.

Aim for 8 to 12 weeks of steady learning before you start applying.

Step 2: build proof of work

You need at least one, ideally two, small projects that someone can open and use.

- Put code on GitHub with a clear README.
- Host web projects on a free hosting service and add the live link.
- For data projects, share a notebook with clean charts and three to five clear findings.
- For design, publish one case study explaining the problem, your process and the final screens.

One complete, polished project beats five half-finished ones.

Step 3: make a one-page resume

- **Header:** name, phone, email, LinkedIn and GitHub or portfolio link.
- **Education:** college, branch, year and current CGPA.
- **Skills:** only what you have used in a project.
- **Projects:** two or three, each with one line on what it does and one line on what you built, plus the link.
- **Activities:** clubs, contests, hackathons or volunteering, if any.

Keep the design simple. Many companies use applicant tracking systems that read plain layouts better.

Step 4: know where to apply

Where	Why it works for 2nd-year students
Faculty research projects	Professors often need help with code, data or experiments and know you already
College incubator startups	Small teams, close to campus, open to students
Startups in your city	Need extra hands and value attitude and basic skill
AICTE internship portal	Lists internships from many organisations for students
LinkedIn and job portals	Filter by "internship" and your skill
Alumni and seniors	A referral gets your resume actually read
Open source programmes	Real code review from experienced developers

Step 5: send good cold emails

A cold email is a short, polite message to someone who has not posted a job. Keep it to five or six lines:

1. Who you are: name, college, branch and year.
2. Why them: one specific line about their product or work.
3. What you can do: your skill and one project link.
4. What you ask for: a short internship, remote or on site, for specific months.
5. Thanks, and your resume attached.

Send it to a founder, team lead or professor directly, not to a generic address if you can avoid it. Follow up once after a week. Do not send the same message to fifty people with only the name changed; it shows.

Month-by-month plan

Months 1 to 2 (start of 3rd semester or earlier): pick a skill, learn the basics, build small exercises.

Month 3: build project 1 and host it.

Month 4: build project 2 or improve project 1 with feedback. Make your resume and LinkedIn profile.

Month 5: apply widely. Talk to two or three professors about research work. Send 5 to 10 personalised cold emails a week.

Month 6: follow up, prepare for short interviews and small test tasks, and accept a suitable offer.

Sample weekly plan while applying

Day	Task
Monday	5 applications on portals
Tuesday	2 personalised cold emails, 1 hour of skill practice
Wednesday	Project improvement
Thursday	5 applications, follow-ups from last week
Friday	Interview practice: explain your project in 2 minutes
Saturday	Learn one new feature or concept, 2 hours
Sunday	Update your tracker and plan the week

How to do well in a short interview

Startups and professors usually ask simple, practical questions: what you built, why you chose a tool, what was hard and how you solved it. Some give a small task to finish in a day or two. Do it carefully, explain your choices in a short note and submit on time. Be honest about what you do not know yet and show that you learn fast.

Making the most of the internship once you get it

- Ask your mentor on day one what "done" looks like for your first task.
- Write a short daily note: what you did, what blocked you, what you will do next. Share it weekly.
- Ask questions after trying for a reasonable time, and bring what you tried.
- Before you leave, ask for a short written reference or a LinkedIn recommendation, and note the project details for your resume.

Avoid fake internships

Be careful if an "internship" asks you to pay a registration or training fee, promises a certificate without any work, or has no clear company website and contact person. Check the company on LinkedIn and ask seniors. Unpaid internships with real work can be acceptable for learning, but paying to work is not an internship.

Checklist before you apply

- [ ] One skill at a basic working level
- [ ] One or two projects with GitHub and live links
- [ ] One-page resume, checked by a senior
- [ ] LinkedIn profile with photo, headline and projects
- [ ] A list of 30 target companies, startups and professors
- [ ] A tracker sheet for applications and follow-ups

If web development is your chosen skill, Uncodemy's Full Stack Development course covers front end, back end and project work.

Frequently asked questions

Can I get an internship in B.Tech 2nd year?

Yes. Startups, faculty research groups, college incubators and NGOs often take 2nd-year students who can show one practical skill and a small project.

Which skill is easiest for a 2nd-year internship?

Web development, data analysis, testing and UI/UX design are common entry points because you can show a project quickly. Choose the one you enjoy.

Do I need a good CGPA for a 2nd-year internship?

It helps, but small companies and professors usually care more about what you can do. Keep your CGPA steady anyway, since it matters later in placements.

Is an unpaid internship worth it in 2nd year?

It can be, if the work is real and you learn something you can put on your resume. Never pay a fee to join an internship.

How many applications should I send?

There is no fixed number. Apply steadily every week, personalise your messages and follow up once. Quality of applications matters more than a big count.

When should I start applying for a summer internship?

Start about two to three months before you want to begin, and earlier for larger companies, which often open applications months in advance.

Is this PDF free?

Yes. Fill in the short form on this page and the PDF downloads right away.

Sources and further reading

- AICTE Internship Portal
- National Career Service portal (Government of India)
- GitHub Docs: about READMEs
- MDN Web Docs: learn web development

About this PDF

Prepared by the Uncodemy career team from the public sources listed. We review it every January. Corrections: info@uncodemy.com.

More free resources and courses: uncodemy.com | info@uncodemy.com | +91 98183 66550