Template from www.lineupx.com/job-descriptions/rpa-developer. Replace every highlighted [field] and keep the version you need: the full-time job for freshers or the internship.

RPA developer (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 RPA developer for the [Team] team. You will build software robots that carry out rule-based work on screens, files and email, starting from a process document agreed with the process owner and finishing with a tested bot and a run note. A senior reviews each bot, and you will learn our tool, our queues and how we fix a bot when a screen changes. The role suits someone patient who likes turning a routine into clear steps.

What you will do

- Review the process definition document with the process owner, list its steps, rules, inputs and exceptions, and put every unclear point to the owner before you build
- Build workflows in [UiPath / Automation Anywhere / Power Automate for desktop] that open applications, type and click through the screen, loop over data tables and run [at the process owner's desk when they start the bot / unattended on a schedule]
- Write selectors, or the tool's equivalent, that find screen elements by stable attributes such as names and ids and not by screen position, and check them again whenever the application changes
- Handle exceptions using the team's pattern: send bad data or broken business rules to a person with the reason. For technical failures, check whether the previous action completed before using a limited retry. Stop and escalate when the outcome is unclear, and test that retries do not create duplicate entries
- Read and write Excel files, CSV files and email attachments, check column names and blank cells before processing, and list every row the bot could not process
- Take credentials from the team's approved credential store when the bot runs, and keep them out of workflow files, spreadsheets, logs and screenshots
- Test each bot in a test environment with data the process owner has approved, covering a normal case, a bad-data case and an application-failure case, and record the results for sign-off
- Log every item the bot handles with its id, the step, the result and the reason, keeping personal data out of logs and screenshots, and write a run note for each bot that says what it does, when it runs and what to check first when it fails
- Check work queues and scheduled runs where the role includes them, under a senior's supervision; list failed items with reasons for the process owner and ask a senior before changing a schedule
- Fix a bot when a screen, a button or a file layout changes: find the failed step in the log, update the selector or the mapping, retest on sample data and record the change in the run note

What we are looking for

- Logic and flow-chart thinking: you can turn a written process into steps, decisions and loops, and draw it as a flow chart
- One programming language at a basic level (variables, conditions, loops and functions), such as Python, Java, C# or VB.NET

- Working knowledge of Excel filters, lookups and simple formulas, with the ability to organise CSV and Excel files and handle email attachments
- Careful testing habits: you try a normal case, a bad-data case and a failure case, and write down what you tried
- Patience with repeated checks, because a bot is run again and again on sample data before the process owner signs it off
- One automation you built and can explain step by step, in an RPA tool, a script or a macro, or described in words if the work is private
- Clear written English for run notes and for questions to the process owner

Good to have

- Practice in the tool named in the post, from a course or a personal project; a vendor certificate is welcome but not needed
- Regular expressions, for pulling values such as invoice numbers out of text
- Excel macros (VBA) or Power Query, for steps that are easier outside the bot
- Basic HTML, so that web selectors built on ids, names and classes make sense
- Calling a REST API from a bot and reading the JSON that comes back, where an application offers one
- Basic SQL for approved read-only checks, such as SELECT with WHERE and JOIN; learn any database-writing steps through the team's reviewed workflow

Who can apply

Final-year students and graduates of any stream can apply, including B.E./B.Tech (computer science, IT, electronics or a related branch), BCA, B.Sc, MCA and BCom. Judge every applicant on the same skills: step-by-step thinking, basic coding, Excel and careful testing.

How to apply

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

RPA developer 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

Build a software robot at [Company name] in this RPA internship. You will automate a practice process with fictional sample data while a mentor reviews each step, and learn how bots are tested, logged and fixed. Interns work on practice processes and test systems, not live records. Machine and tool access: [state what [Company name] provides]. Using your work in a portfolio, or publishing it, needs approval from [Company name].

What you will do

- Draw a flow chart for a practice process from its written description, and list the questions you would put to the process owner
- Build one bot for a practice process, such as copying rows from a sample Excel file into a practice web form, and run it on fictional sample data
- Add handling for a missing file, a blank field and a page that does not load, and show how the bot records each one
- Write test cases with the expected result for each, and record the outcome of every run
- Watch the bot fail after your mentor changes the practice screen, then fix the selector and note what changed
- Write a one-page run note covering what the bot does, its inputs and outputs and what to check first if it fails, and present it with the bot to the team

What we are looking for

- A current student or recent graduate of any stream
- Can draw a flow chart for an everyday process with decisions and loops, such as checking a library return or an order
- Knows the basics of one programming language: variables, conditions and loops
- Uses Excel for sorting, filtering and simple formulas
- Has tried automating a small task with an RPA tool, a script or a macro, even from a course, and can describe what it did

Who can apply

Final-year students and graduates of any stream can apply, including B.E./B.Tech (computer science, IT, electronics or a related branch), BCA, B.Sc, MCA and BCom. Judge every applicant on the same skills: step-by-step thinking, basic coding, Excel and careful testing.

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. **A bot that ran well last week now fails because a button has moved or been renamed. What do you check?** Look for the log and the failed step as the starting point, then a comparison of the element on screen with the stored selector, and checks for a new application version, a pop-up, slow loading or a different screen scale. Accept the tool's own word for a selector.
2. **A bot can fail because an item has bad data, or because the application has a technical fault. How should the bot treat each? Give an example of each.** Look for separating bad data or a broken business rule, which needs correction or a person's decision, from a technical failure that may allow a limited retry. Before retrying, check whether the previous action completed; stop and escalate if the outcome is unclear. Accept the candidate's own words and examples instead of vendor terminology.
3. **How should a bot get the password it needs to log in to an application?** Look for using the team's approved authentication method and keeping passwords out of workflow files, spreadsheets, logs and screenshots. For an attended bot, credentials and permissions must stay within the signed-in user's authorised access. For an unattended bot, accept credentials retrieved from an approved store under a senior's configuration. No vendor-specific terminology is needed.
4. **How would you test a bot that posts entries into another system before it touches real records?** Look for a test environment or a dry run, sample rows that include bad data, a check that running the bot twice does not post an entry twice, and a count of rows read against entries made. Accept equivalent checks in the candidate's own tool.
5. **A process document says 'then use your judgement' for one step. What do you do?** Look for asking the process owner for the rule behind the judgement, or leaving that step with a person, instead of guessing inside the bot. Accept a candidate who asks for examples of past decisions.
6. **A bot reads an Excel file every morning, and today a column has been renamed. What should the bot do?** Look for a check of the column names before processing, a clear message in the log, a stop or a hand-off to a person instead of reading the wrong column, and a note to the file's owner. Accept checks on column order or row count as well.
7. **Describe one task you automated with an RPA tool, a script or a macro. What did it do when the input was wrong?** Look for a real task, the steps in order and one handled failure. Accept a course project, a spreadsheet macro or private work described in words; no link or particular tool is needed.