



Software Engineering
IT314: Final Submission
Group 22
Unit Testing

1. Login Module (Method is POST, Route for this module is '/profile')

- Logging in through valid credentials

 A screenshot of the Postman application interface. On the left, there's a sidebar with 'Team Workspace' containing 'Collections' (with 'Login'), 'Environments', and 'History'. The main area shows a 'POST LoginStudent' request. The 'Body' tab is selected, showing three parameters: 'email' (202001226@daii.ac.in), 'password' (Jatin@123), and 'role' (Student). Below the body, the response section shows a status of '200 OK' with a green checkmark, a response time of '244 ms', and a size of '4.62 KB'. The response content is displayed in 'Pretty' format as an HTML document.


```

1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <title>Student Profile</title>
6   <meta charset="utf-8">
7   <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
8
9   <link href="https://fonts.googleapis.com/css?family=Poppins:300,400,500,600,700,800,900" rel="stylesheet">
10  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.4.0/css/all.min.css"
11    integrity="sha512-iecdLmaskI7CVkqXN0/ZH/XlWZOJyj7YY7tcenmpA5oypASozpmT/E0iPtmFIB46ZmdtAc9eNBvHOH/ZpiBw=="
12    crossorigin="anonymous" referrerPolicy="no-referrer" />
13  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.4.0/css/brands.min.css"
14    integrity="sha512-9YHSK59/rjvhTcY/b4xdl0v4LPDwdkce0182LF5TB6745ml1Af1uEU6dFWqwDw91PvnauxFgpKvJqp7jiQ==" crossorigin="anonymous" referrerPolicy="no-referrer" />
15  <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css">
16  <link rel="stylesheet" href="/css/style.css">
17
  
```

As we can see from the above screenshot, route `/profile` is returning HTML file means it will be displayed after this post request. It will login the requested post request data into the system.

It also set the cookies inside the browser as we can see it coming in the response of this POST request.

The screenshot shows the Postman interface with the 'Cookies' tab selected. A single cookie is listed:

Name	Value	Domain	Path	Expires	HttpOnly	Secure
jwt	eyJhbGciOiJI... (redacted)	localhost	/	Wed, 26 Apr 1... (redacted)	true	false

The status bar at the top indicates: Status: 200 OK Time: 244 ms Size: 4.62 KB. Below the main table are standard Postman navigation and search buttons.

- Logging in through invalid credentials (password)

The screenshot shows the Postman interface with the 'Body' tab selected. A POST request is made to `http://localhost:3000/profile`. The 'Body' section contains the following parameters:

Key	Value	Description	... Bulk Edit
email	202001226@daiict.ac.in		
password	Jatin@12345		
role	Student		

The response status is 500 Internal Server Error. The response body is a HTML page with the message 'Password Invalid'.

```

1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta http-equiv="X-UA-Compatible" content="IE=edge">
7   <meta name="viewport" content="width=device-width, initial-scale=1.0">
8   <link rel="stylesheet" href="/css/404-2.css">
9   <link rel="stylesheet" href="/css/profile.css">
10  <title>404</title>
11 </head>
12
13 <body>
14
15   <div class="container">
16     
17
18   <div class="error1"> Password Invalid </div>
19   <a href="/">
20     <div class="home"> Back to Home </div>

```

As we can see from the above POST request's response, It is returning the error page with the message of 'Password Invalid'. Here in continuing with the previous request which was successful, we can see that this request is fail still it is sending us the cookies. Reason behind it is we are trying to login into another account but we didn't succeeded. So we remains logged in, in the previous account.

- Logging in through wrong role

The screenshot shows the Postman interface with a POST request to `http://localhost:3000/profile`. The request body contains three parameters: `email` (value: `202001226@daiict.ac.in`), `password` (value: `Jatin@123`), and `role` (value: `Admin`). The response status is `500 Internal Server Error`, indicating that the user does not exist in the database.

```
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta http-equiv="X-UA-Compatible" content="IE=edge">
7   <meta name="viewport" content="width=device-width, initial-scale=1.0">
8   <link rel="stylesheet" href="/css/404-2.css">
9   <link rel="stylesheet" href="/css/profile.css">
10  <title>404</title>
11 </head>
12
13 <body>
14
15  <div class="container">
16    
17
18    <div class="error1"> Error retrieving user with email 202001226@daiict.ac.in </div>
19    <a href="/">
20      <div class="home"> Back to Home </div>
```

In our implementation, we are finding the users based on the role given. Obviously student is not existing in the admin database. Because of that, our post request to '/profile' route returning the error page with the message that user doesn't not exist in admin database.

- Logging in through wrong email

The screenshot shows the Postman interface with a POST request to `http://localhost:3000/profile`. The request body contains three parameters: `email` (value: `202001227@daiict.ac.in`), `password` (value: `Jatin@123`), and `role` (value: `Student`). The response status is `500 Internal Server Error`, indicating that the user does not exist in the database.

```
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta http-equiv="X-UA-Compatible" content="IE=edge">
7   <meta name="viewport" content="width=device-width, initial-scale=1.0">
8   <link rel="stylesheet" href="/css/404-2.css">
9   <link rel="stylesheet" href="/css/profile.css">
10  <title>404</title>
11 </head>
12
13 <body>
14
15  <div class="container">
16    
17
18    <div class="error1"> Error retrieving user with email 202001227@daiict.ac.in </div>
19    <a href="/">
20      <div class="home"> Back to Home </div>
```

Now, role is 'Student' but we are logging through the email which is not yet registered. So it is giving the response like 'Error retrieving user with email ...' This works in the same way for all the roles in our system including companies and admins.

We have also tested this route in the lab session using chai and mocha frameworks of Node.js and result of it is shown below.

The screenshot shows a VS Code interface with the following details:

- EXPLORER** view: Shows the project structure with files like controller.js, server.js, and login.js.
- TERMINAL** view:
 - Output of the command: `apple@jatin1510 Placement Cell % npm test`
 - Output of the command: `> placementcell@1.0.0 test`
 - Output of the command: `> mocha test/login.js`
 - Test results:
 - Testing route /login
 - 18 passing (6s)

The terminal output shows 18 passing tests, indicating successful execution of the test cases for the /login route.

2. Companies Filter Module for Student (Method is GET, Route for this module is '/viewCompany')

Name	Criteria	CTC	Open For	Job Title	Details	(De)Register
Google	7	45 LPA	M.Tech , MSc	Software Engineer	Details	Register
Google	8	60 LPA	B.Tech , M.Tech	Hardware Developer	Details	Register
Google	7	22 LPA	MSc	Communication Engineer	Details	Branch
Microsoft	8.5	36 LPA	B.Tech , M.Tech , MSc	Azure Developer	Details	CPI
Microsoft	6	19 LPA	B.Tech , MSc	Microsoft 365 Developer	Details	Branch
Wipro	6	9 LPA	B.Tech , M.Tech , MSc	Full stack developer	Details	Register
tekion	7	1500000 LPA	B.Tech	tekion	Details	Branch
testing firm	9.8	60 LPA	B.Tech	job-1	Details	CPI
testing firm	8	40 LPA	M.Tech	job-2	Details	Not Started

Expected: Student can see the companies in which (s)he can apply.

As we can see from the above GET request, request is serving the page in which student easily register in the companies via cpi criteria, branch criteria.

- Student wants to see companies which is coming for his/her branch

Name	Criteria	CTC	Open For	Job Title	Details	(De)Register
Google	8	60 LPA	B.Tech , M.Tech	Hardware Developer	Details	Register
Microsoft	8.5	36 LPA	B.Tech , M.Tech , MSc	Azure Developer	Details	CPI
Microsoft	6	19 LPA	B.Tech , MSc	Microsoft 365 Developer	Details	Register
Wipro	6	9 LPA	B.Tech , M.Tech , MSc	Full stack developer	Details	Register
tekion	7	1500000 LPA	B.Tech	tekion	Details	Not Started
testing firm	9.8	60 LPA	B.Tech	job-1	Details	CPI
Twitter	4	20 LPA	B.Tech	SDE	Details	Not Started
acer	6	15 LPA	B.Tech , MSc	developer	Details	Not Started

Expected: Students can only see the jobs which is coming for his/her branch.

As we can see from the above POST request, students can see all the jobs (which is of particular companies) which is coming for placement to their branch.

Based on the input given in the form, or in the POST method, student can always apply various filters on the jobs of companies.

3. Update Password Module

Test cases:

1. Role Student: Testing route /studentUpdatePassword

Wrong Role: Status should be 500

Confirm password not matching: Status should be 400

Confirm password matching, OLD Password Wrong: Status should be 400

All parameters are right: Status should be 200

2. Role Admin: Testing route /adminUpdatePassword

Wrong Role: Status should be 500

Confirm password not matching: Status should be 400

Confirm password matching, OLD Password Wrong: Status should be 400

All parameters are right: Status should be 200

3. Role Company: Testing route /companyUpdatePassword

Wrong Role: Status should be 500

Confirm password not matching: Status should be 400

Confirm password matching, OLD Password Wrong: Status should be 400

All parameters are right: Status should be 200

4. Role Super Admin: Testing route /adminUpdatePassword

Wrong Role: Status should be 500

Confirm password not matching: Status should be 400

Confirm password matching, OLD Password Wrong: Status should be 400

All parameters are right: Status should be 200

Output:

The screenshot shows the VS Code interface with the following details:

- EXPLORER**: Shows the project structure:
 - OPEN EDITORS: router.js, controller.js M, UpdatePasswordTest.js U, LoginTest.js U, package.json M.
 - PLACEMENT-CELL: assets, node_modules, server (controller.js M), database, middleware (middleware.js), model (model.js), routes (router.js), service, test (LoginTest.js, UpdatePasswordTest.js U), views, .vscode (settings.json).
- TERMINAL**:
 - Test command: `test > UpdatePasswordTest.js > ...`
 - Code snippet:

```
const host = "http://localhost:5000";
describe('Role Student: Testing route /studentUpdatePassword', () =>
{
  const path = "/studentUpdatePassword";
```
 - npm test command:

```
apple@jatin1510 Placement-Cell % npm test
```
 - Test output:
 - Role Student: Testing route /studentUpdatePassword
 - Wrong Role: Status should be 500
 - Confirm password not matching: Status should be 400
 - Confirm password matching, OLD Password Wrong: Status should be 400
 - All parameters are right: Status should be 200
 - Role Admin: Testing route /adminUpdatePassword
 - Wrong Role: Status should be 500
 - Confirm password not matching: Status should be 400
 - Confirm password matching, OLD Password Wrong: Status should be 400
 - All parameters are right: Status should be 200
 - Role Company: Testing route /companyUpdatePassword
 - Wrong Role: Status should be 500
 - Confirm password not matching: Status should be 400
 - Confirm password matching, OLD Password Wrong: Status should be 400
 - All parameters are right: Status should be 200
 - Role SuperAdmin: Testing route /adminUpdatePassword
 - Wrong Role: Status should be 500
 - Confirm password not matching: Status should be 400
 - Confirm password matching, OLD Password Wrong: Status should be 400
 - All parameters are right: Status should be 200
 - Test summary:

```
16 passing (6s)
```
- OUTPUT**: Shows the terminal output from the npm test command.
- PROBLEMS**: Shows no problems.
- TRUFFLE**: Shows no truffle tests.
- COMMENTS**: Shows no comments.
- DEBUG CONSOLE**: Shows no debug console output.
- SIDE BAR**: Shows node and zsh options.

Admin Testing:

1) Without Login

First of all , Admin need to login into page.

The screenshot shows the Postman application interface. On the left, the sidebar includes sections for Collections, APIs, Environments, and History. A 'Well done! You got the basics.' message is displayed with an icon of a trophy. The main workspace shows an 'Admin' collection with a 'New Collection' entry. A failed API request is shown for 'http://localhost:3000/datasheet' with a status of 500 Internal Server Error. The response body is a 404 error page from the '404s' collection. A 'Got it!' button is at the bottom left.

Well done!
You got the basics.

Postman also enables you to automate API testing, monitor API performance, mock APIs and more.

Try these with our interactive Bootcamp lessons.

Got it!

From above , it also shows that access denied. That means first need to login as an admin , then only you can access insider path(routes).

2) Login

Then After we have logged in as a Placement Manager.

The screenshot shows the Postman interface with a successful POST request to `http://localhost:3000/profile`. The request body includes:

email	placement@dailct.ac.in
password	superadmin
role	Placement Manager

The response body shows the rendered HTML of a profile page for a Placement Manager, including navigation links for Admin Statistics, Update Password, Verify Students, Schedule Jobs, and Verify Jobs.

As we can see here , it render profile page of Placement Manager. Which is expected.

3) Check For Students whose verification is pending.

The screenshot shows the Postman interface with a successful GET request to `http://localhost:3000/unverifiedStudents`. The response body shows the rendered HTML of a page listing unverified students, with links for Verify Students, Schedule Jobs, and Verify Jobs.

Here Admin Requested for list of students whose verification is pending. For given API GET request , it will redirect to 'Verify Students' page.

4) Check For Jobs which Verification is pending.

First of all unknown API request leads to error message.

The screenshot shows the Postman interface with a collection named 'Admin'. A new request is being created for the URL `http://localhost:3000/unverifiedJobs`. The 'Body' tab is selected, showing a raw HTML response:

```

1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="utf-8">
6   <title>Error</title>
7 </head>
8
9 <body>
10  <pre>Cannot GET /unverifiedJobs</pre>
11 </body>
12
13 </html>

```

The status bar at the bottom indicates a 404 Not Found error with a time of 7 ms and a size of 425 B.

Let us look at preview for requested GET API request. Here it shows page without CSS.

The screenshot shows the same Postman interface with the same setup. The status bar now indicates a 200 OK response with a time of 256 ms and a size of 4.92 KB. The 'Preview' tab is selected, showing a list of verify jobs:

- ✓ Verify Students
- ✓ Schedule Jobs
- ✓ Verify Jobs
- ✓ Verify Company
- ☰ Student list
- ☰ Job list
- ☰ Company list
- ☰ Datasheet
- ✖ Logout

Below this, a 'Verify Jobs' section is shown with a table:

Company name	Job name	CPI criteria	CTC	Accept	Reject
Google	Communication Engineer	7	22	<button>Accept</button>	<button>Reject</button>
Microsoft	Developer	5	26	<button>Accept</button>	<button>Reject</button>

5) Check For Company which verification is pending.

The screenshot shows the Postman interface with a GET request to `http://localhost:3000/unverifiedCompany`. The response body contains a list of companies:

- Wells Fargo
- j
- D.E.shaw
- google
- acer

For each company, there are 'Accept' and 'Reject' buttons. The status bar at the bottom indicates `Status: 200 OK Time: 253 ms Size: 8.04 KB`.

Same as Jobs and Students, we can see list of all companies which verification is pending. Here we have logged in as a Placement Manager, that's why it allows to verify Company.

In below screenshot, user has logged in as a “Admin”, and tried to access list of unverified companies. But as we know only super Admin (Placement Manager) can access list.

So from below we can see that, system don't allow to access.

The screenshot shows the Postman interface with a POST request to `http://localhost:3000/unverifiedCompany`. The response status is `404 Not Found`. The response body is:

```

1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="utf-8">
6   <title>Error</title>
7 </head>
8
9 <body>
10  <pre>Cannot POST /unverifiedCompany</pre>
11 </body>
12
13 </html>

```

6) Accept Request Call For Company.

Request:

<http://localhost:3000/verifyCompany/6447e8426f3e7e34698259d6>

6447e8426f3e7e34698259d6 is object id of company wells-Fargo. We pass this id via “params” and using that we can know for which company verification request has been called.

In point 5, screenshot we verification request for wells-Fargo is in pending list but after generating API request for verification, from below screenshot we can see Wells Fargo has been gone from pending list.

The screenshot shows the Postman interface with a successful API call. The URL is `http://localhost:3000/verifyCompany/6447e8426f3e7e34698259d6`. The response body contains a list of items under the heading "Registered company".

Company Name	Email	HR Name	Website	Contact number	Accept	Reject
j	fg@fg	th	Click Here	55555-55555	Accept	Reject
D.E.shaw	kuldeepprajapati3902@gmail.com	Mr Been	Click Here	95584-26978	Accept	Reject
google	Anurag@gmail.com	anurag	Click Here	12345-01234	Accept	Reject
acer	anuragjanv@gmail.com	anurag	Click Here	12345-01234	Accept	Reject

7) Display List of all Students, Company, Jobs.

Students:

The screenshot shows the Postman interface with a successful API call. The URL is `http://localhost:3000/adminStudents`. The response body contains a list of students under the heading "Placement Manager".

Nikhil Vaghasiya
Viraj Pansuriya
Jaimin Satani
Hiten Mistry
Kathan Shah

It will list of all students which have registered for placement Drive. Same as for Jobs and Company We can see below.

Jobs:

The screenshot shows the Postman interface with a successful API call to `http://localhost:3000/adminJobs`. The response body displays a list of student records under the heading "Placement Manager".

	Value
Google : Software Engineer	✓
Google : Hardware Developer	✓
Google : Communication Engineer	✓
Microsoft : Azure Developer	✓
Microsoft : Microsoft 365 Developer	✓

Company:

The screenshot shows the Postman interface with a successful API call to `http://localhost:3000/adminCompany`. The response body displays a list of company records under the heading "Placement Manager".

	Value
Google	✓
Microsoft	✓
Morgan Stanley	✓
Wells Fargo	✓
tekion	✓

8) Accept Request for Student and Job.

For Student:

Before verification Request:

Well done! You got the basics.

Postman also enables you to automate API testing, monitor API performance, mock APIs and more. Try these with our interactive Bootcamp lessons. Got It

GET http://localhost:3000/unverifiedstudents

Params Authorization Headers (9) Body Pre-request Script Tests Settings Cookies

Query Params

Key	Value	Description	Bulk Edit
Key	Value	Description	(i)

Body Cookies (1) Headers (7) Test Results

Pretty Raw Preview Visualize

Status: 200 OK Time: 231 ms Size: 5.3 KB Save as Example

• Verify Students
• Schedule Jobs
• Verify Jobs
• Verify Company
• Student list
• Job list
• Company list
• Datasheet
• Logout

Registered Students

Name	Branch	CPI	Current Backlog(s)	Total Backlog(s)	Accept	Reject
Dev	M.Tech	8.3	0	0	<button>Accept</button>	<button>Reject</button>
Jatin	B.Tech	8.16	0	0	<button>Accept</button>	<button>Reject</button>

After Verification Request:

Well done! You got the basics.

Postman also enables you to automate API testing, monitor API performance, mock APIs and more. Try these with our interactive Bootcamp lessons. Got It

GET http://localhost:3000/verifyStudent/64480bad2e7e3f58b138f0b0

Params Authorization Headers (9) Body Pre-request Script Tests Settings Cookies

Query Params

Key	Value	Description	Bulk Edit
Key	Value	Description	(i)

Body Cookies (1) Headers (7) Test Results

Pretty Raw Preview Visualize

Status: 200 OK Time: 229 ms Size: 4.73 KB Save as Example

• Update Password
• Verify Students
• Schedule Jobs
• Verify Jobs
• Verify Company
• Student list
• Job list
• Company list
• Datasheet
• Logout

Registered Students

Name	Branch	CPI	Current Backlog(s)	Total Backlog(s)	Accept	Reject
Jatin	B.Tech	8.16	0	0	<button>Accept</button>	<button>Reject</button>

From above Two screenshot, we can see that after verification GET request, “DEV” has been disappeared from verification pending student list. Which Shows that Dev has been successfully verified.

Same for Job also,

Before Verification Request:

The screenshot shows the Postman interface with a successful API request. The request URL is `http://localhost:3000/unverifiedjobs`. The response body contains a list of jobs:

- ✓ Verify Students
- ✗ Schedule Jobs
- ✗ Verify Jobs
- ✗ Verify Company
- ✗ Student list
- ✗ Job list
- ✗ Company list
- ✗ Datasheet
- ✗ Logout

Below the list, there is a table titled "Verify Jobs" showing details for Google's Communications Engineer job:

Company name	Job name	CPI criteria CTC	Accept	Reject
Google	Communication Engineer 7	22	Accept	Reject
Microsoft	Developer	5	Accept	Reject

After Verification for Google's Communications Engineer Job:

The screenshot shows the Postman interface with a successful API request. The request URL is `http://localhost:3000/verifyjob/644786440f794511d7a4dc13`. The response body contains a list of jobs, but Google's Communications Engineer job has been removed:

- ✓ Update Password
- ✓ Verify Students
- ✗ Schedule Jobs
- ✗ Verify Jobs
- ✗ Verify Company
- ✗ Student list
- ✗ Job list
- ✗ Company list
- ✗ Datasheet
- ✗ Logout

Below the list, there is a table titled "Verify Jobs" showing details for Microsoft's Developer job:

Company name	Job name	CPI criteria CTC	Accept	Reject
Microsoft	Developer 5	26	Accept	Reject

Same as above, Google has been disappeared.

9) Update Password

Well done! You got the basics.

Postman also enables you to automate API testing, monitor API performance, mock APIs and more.

Try these with our interactive Bootcamp lessons.

Get It

10) Datasheet Download;

/datasheet will generate datasheet for unplaced students which are not rejected by admin , and excel sheet will be automatically downloaded to user's computer.

Well done! You got the basics.

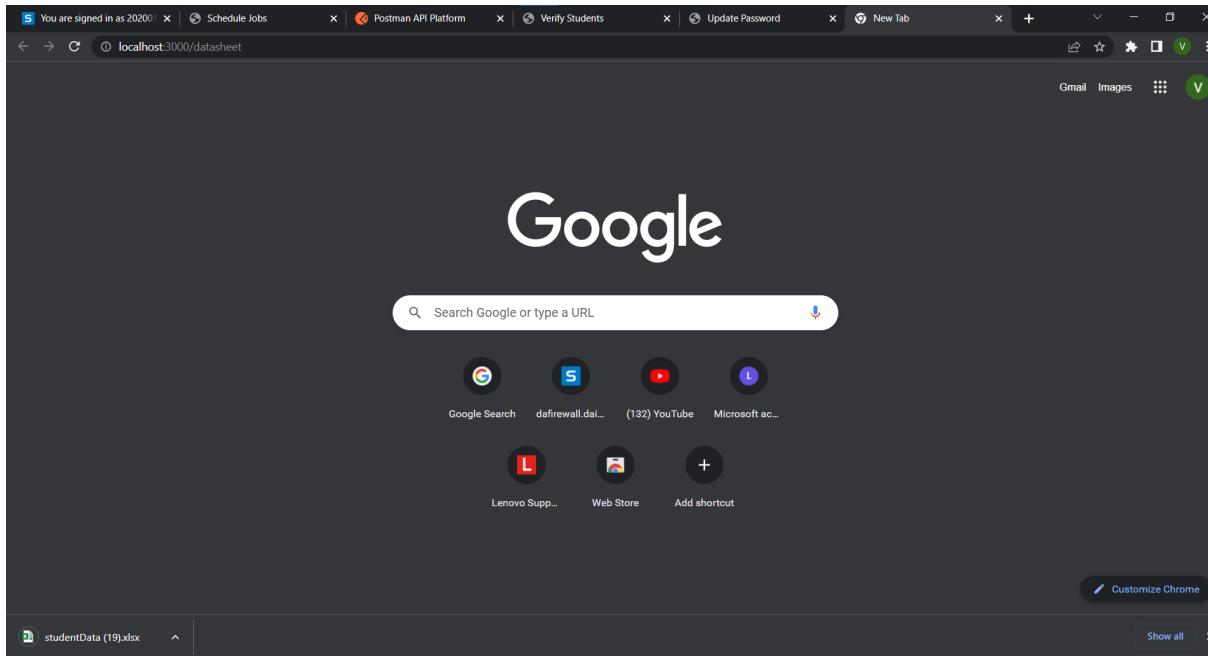
Postman also enables you to automate API testing, monitor API performance, mock APIs and more.

Try these with our interactive Bootcamp lessons.

Get It

Here, from preview file can't understand anything because this request will not return or render anything. It will just download datasheet in user's computer.

Let me run that request in browser,



From above screenshot, we can see that datasheet has been downloaded.

11) Schedule Jobs

Request: <http://localhost:3000/adminInterviewSchedule>

The screenshot shows the Postman application interface. On the left, there is a sidebar with sections for "My Workspace", "Collections", "APIs", "Environments", and "History". A "Well done! You got the basics." message is displayed with a trophy icon.

In the main workspace, there is a collection named "Admin" containing a single GET request to "http://localhost:3000/adminInterviewSchedule". The request has 9 headers. The response body is a table titled "Interview schedule" with the following data:

Company Name	Job	Start Dates	End Dates	Confirm
Google	Software Engineer	24-04-2023	27-04-2023	<input type="checkbox"/> Submit
Google	Hardware Developer	25-04-2023	27-04-2023	<input type="checkbox"/> Submit
Google	Communication Engineer	25-04-2023	25-04-2023	<input type="checkbox"/> Submit
Microsoft	Azure Developer	25-04-2023	27-04-2023	<input type="checkbox"/> Submit
Microsoft	Microsoft 365 Developer	25-04-2023	27-04-2023	<input type="checkbox"/> Submit
Wipro	Full stack developer	25-04-2023	27-04-2023	<input type="checkbox"/> Submit
tekion	tekion	25-04-2023	25-04-2023	<input type="checkbox"/> Submit
testing firm	job-1	25-04-2023	25-04-2023	<input type="checkbox"/> Submit
testing firm	job-2	25-04-2023	25-04-2023	<input type="checkbox"/> Submit

From above screenshot we can see that , it will list up all verified jobs.

12) Rejection Request:

Student:

The screenshot shows the Postman interface with a collection named "Admin". A GET request is made to `http://localhost:3000/unverifiedstudents`. The response body displays a table of registered students with columns: Name, Branch, CPI, Current Backlog(s), Total Backlog(s), Accept, and Reject. The "Accept" button is highlighted in red.

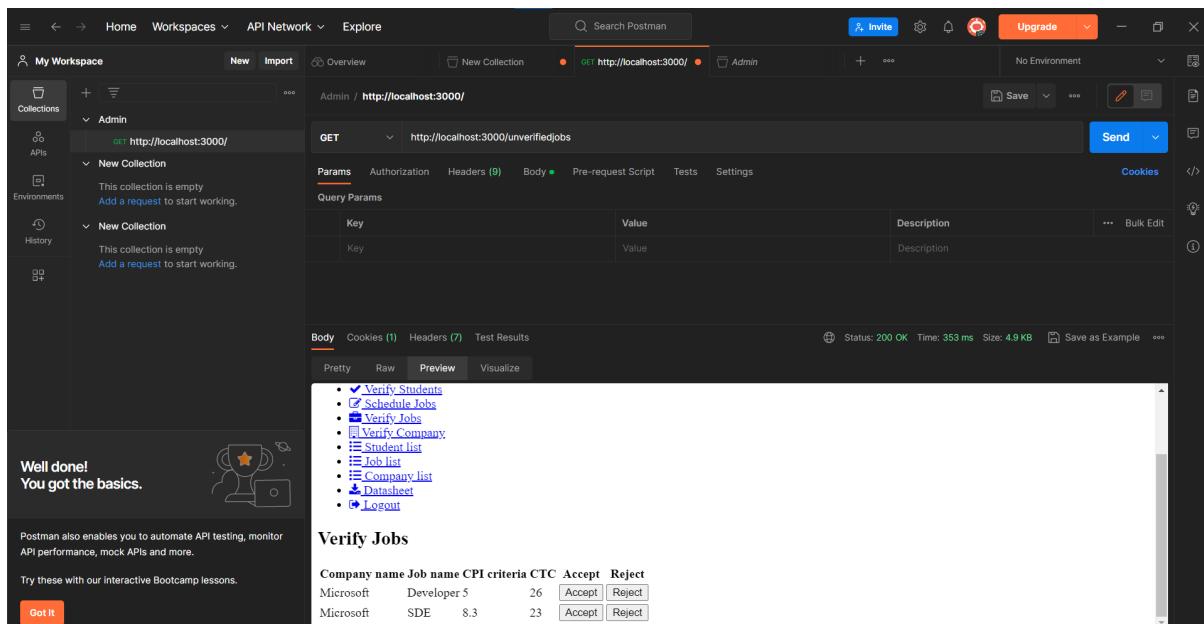
Name	Branch	CPI	Current Backlog(s)	Total Backlog(s)	Accept	Reject
Kathan	B.Tech	7.42	0	0	Accept	Reject
jainam	B.Tech	9	0	0	Accept	Reject
test	B.Tech	10	10	7	Accept	Reject

The screenshot shows the Postman interface with a collection named "Admin". A GET request is made to `http://localhost:3000/rejectStudent/64483885a1c318c02e0bd1d`. The response body displays a table of registered students with columns: Name, Branch, CPI, Current Backlog(s), Total Backlog(s), Accept, and Reject. The "Accept" button is highlighted in red.

Name	Branch	CPI	Current Backlog(s)	Total Backlog(s)	Accept	Reject
jainam	B.Tech	9	0	0	Accept	Reject
test	B.Tech	10	10	7	Accept	Reject

As we have seen for verification request of students , company and Jobs. Here we are testing with Rejection request. Functionality of both request is somewhere about same.

From above two screenshots we can see that first of all Kathan was in pending list, but Rejection request his name has been gone from pending verification list.

Job:**Rejection request of Microsoft's Developer job:**

Well done! You got the basics.

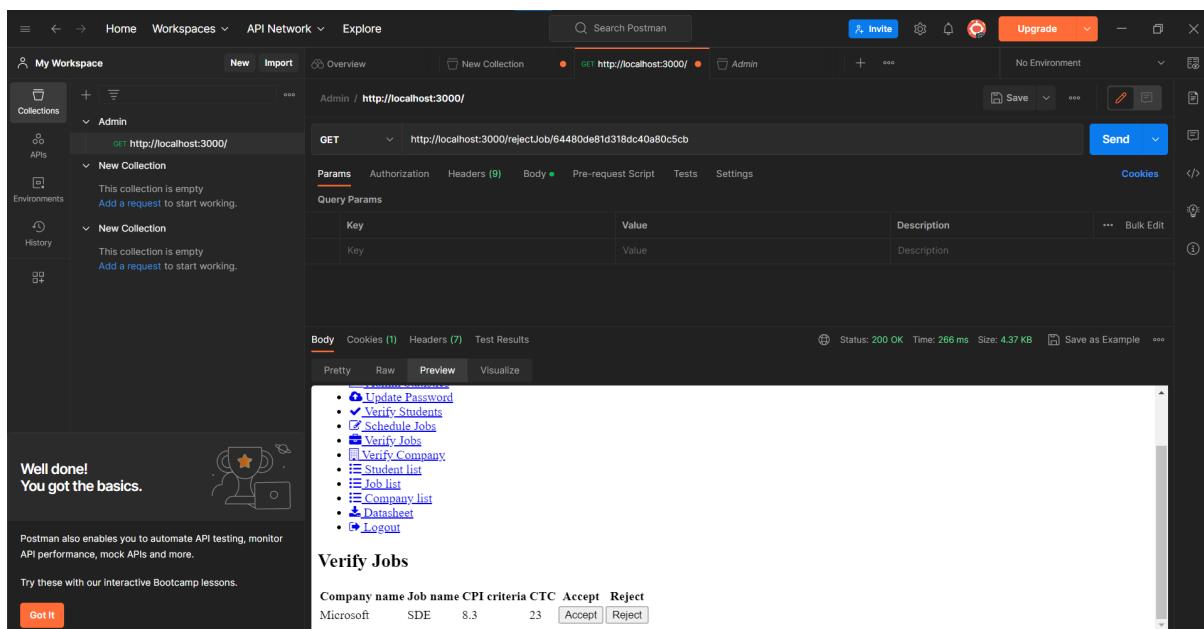
Postman also enables you to automate API testing, monitor API performance, mock APIs and more.

Try these with our interactive Bootcamp lessons.

Got It

Status: 200 OK Time: 353 ms Size: 4.9 KB Save as Example

Company name	Job name	CPI	criteria	CTC	Accept	Reject
Microsoft	Developer	5	26	26	<button>Accept</button>	<button>Reject</button>
Microsoft	SDE	8.3	23	23	<button>Accept</button>	<button>Reject</button>



Well done! You got the basics.

Postman also enables you to automate API testing, monitor API performance, mock APIs and more.

Try these with our interactive Bootcamp lessons.

Got It

Status: 200 OK Time: 266 ms Size: 4.37 KB Save as Example

Company name	Job name	CPI	criteria	CTC	Accept	Reject
Microsoft	SDE	8.3	23	23	<button>Accept</button>	<button>Reject</button>

Company:

My Workspace

Admin / http://localhost:3000/

GET http://localhost:3000/unverifiedcompany

Company Name	Email	HR Name	Website	Contact number	Accept	Reject
Amazon	202001197@daiict.ac.in	Maitri	Click Here	78748-91891	Accept	Reject
Amazon	202001053@daiict.ac.in	Krish	Click Here	95584-26978	Accept	Reject
Google	kathanshah116@daiict.ac.in	Kathan Shah	Click Here	91547-34134	Accept	Reject

My Workspace

Admin / http://localhost:3000/

GET http://localhost:3000/rejectCompany/6448f2d9fc24eab85dd9a96f

Company Name	Email	HR Name	Website	Contact number	Accept	Reject
Amazon	202001197@daiict.ac.in	Maitri	Click Here	78748-91891	Accept	Reject
Amazon	202001053@daiict.ac.in	Krish	Click Here	95584-26978	Accept	Reject
Google	kathanshah116@daiict.ac.in	Kathan Shah	Click Here	91547-34134	Accept	Reject

From above screen shot , we can see that first of all we have requested for all unverified company and it display list of that. Then after we have generated GET request to reject particular company.

Let say here we want to reject “Amazon” which HR is Maitri. But from above Result we can see that even after doing GET request there is no change in pending list.

Here we have found our Mistake.

Mistake: when we merge our code, we forgot to add “isRejected” flag in Company Schema. So in controller.js file which contains all kind of function module

whenever we tried to mark “isRejected” flag as true, we are unable to find that kind of attribute in database. That is reason behind this error.

After Resolving Error:

The screenshot shows the Postman interface with a collection named 'Admin'. A GET request is made to `http://localhost:3000/unverifiedcompany`. The 'Body' tab is selected, showing form-data parameters: email (pans@gmail.com), password (1234), and role (Placement Manager). The response status is 200 OK, time 64 ms, size 5.78 KB. The 'Test Results' section lists various API endpoints like Verify Students, Interview schedule, Verify Jobs, Verify Company, Student list, Job list, Company list, Datasheet, and Logout.

Company Name	Email	HR Name	Website	Contact number	Accept	Reject
sprinklr	sprinklr@google.com	poojan	www.sprinklr.com	9016527877	<button>Accept</button>	<button>Reject</button>
microsoft	microsoft@google.com	jainisha	www.microsoft.com	9913281762	<button>Accept</button>	<button>Reject</button>

The screenshot shows the Postman interface with a collection named 'Admin'. A GET request is made to `http://localhost:3000/rejectCompany/6443adc896667c6542aa2b4e`. The 'Body' tab is selected, showing form-data parameters: email (pans@gmail.com), password (1234), and role (Placement Manager). The response status is 200 OK, time 67 ms, size 5.09 KB. The 'Test Results' section lists various API endpoints.

Company Name	Email	HR Name	Website	Contact number	Accept	Reject
microsoft	microsoft@google.com	jainisha	www.microsoft.com	9913281762	<button>Accept</button>	<button>Reject</button>

After resolving above mentioned error, we can see that it works properly.