

Non-Functional Test Plan for CCR

USABILITY TESTING

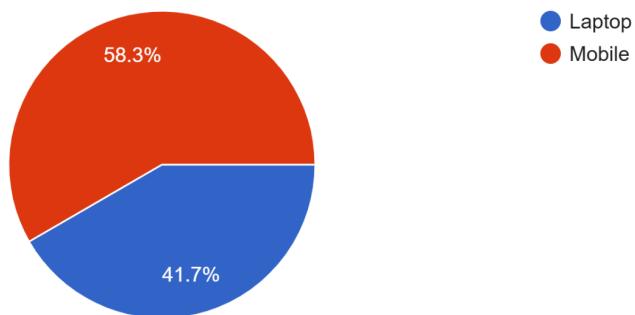
Usability testing checks if **real users can use your system easily.**

- Time taken
- Steps taken
- Where they struggled
- Errors/messages
- UI confusion

SUS Questionnaire (google forms)

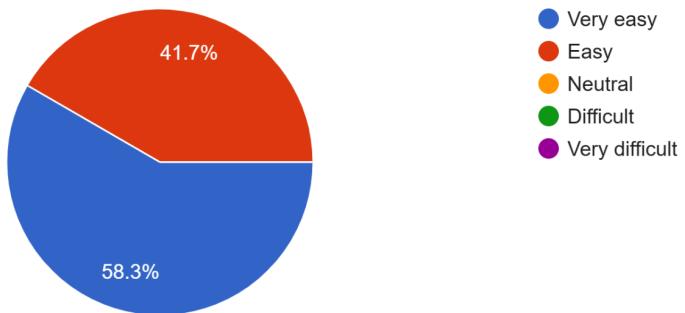
Device Used?

12 responses



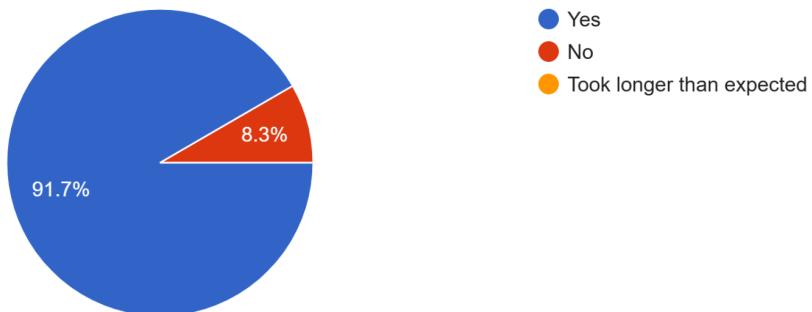
How easy was the login/logout process?

12 responses



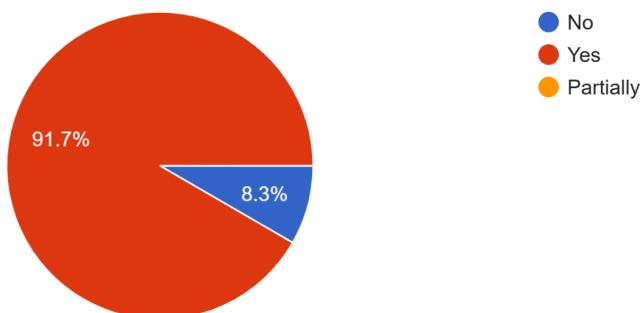
Were you able to find the complaint status easily?

12 responses



Were you able to complete the task? (Submit a Complaint)

12 responses



Time taken to complete (in seconds)

12 responses

10 sec

40

5

2s

45

~60 seconds

300

60

10

Describe any difficulties you faced (confusion or issues)

6 responses

Complaint submision took a little time

None

why did my wifi not working problem go to tech support? i dont think they handle wifi issues. hmc should be informed i guess.

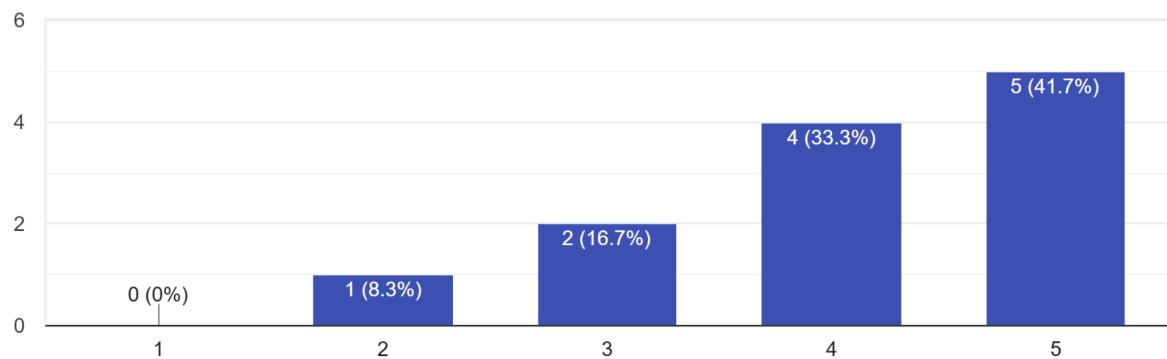
'Submit anonymously' button didn't work

NA

No I didn't faced any technical issue as such

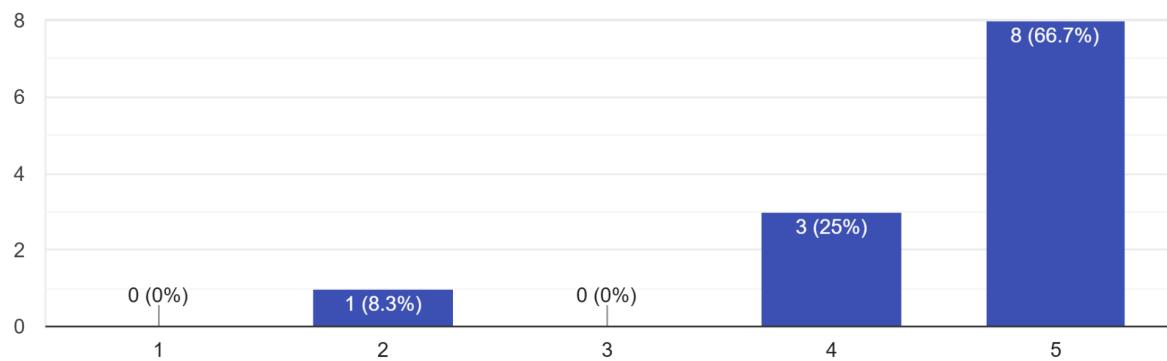
I think I would like to use this system frequently.

12 responses



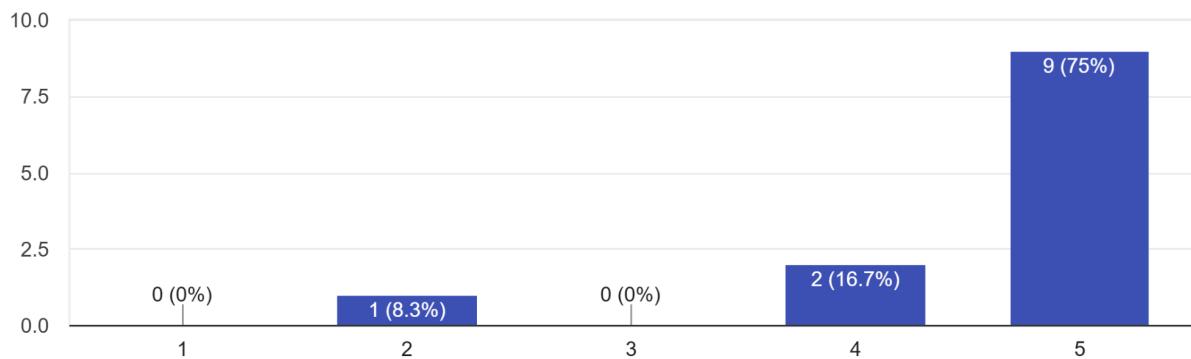
I thought the system was easy to use.

12 responses



I imagine most people would learn to use the system quickly.

12 responses



SECTION 4

What did you like the most about the system?

7 responses

Easy to use

UpVote System.

How easy it is to complain. Nice job.

It's availability

I liked the ease in use of application along with its smooth flow.

It's easy to go

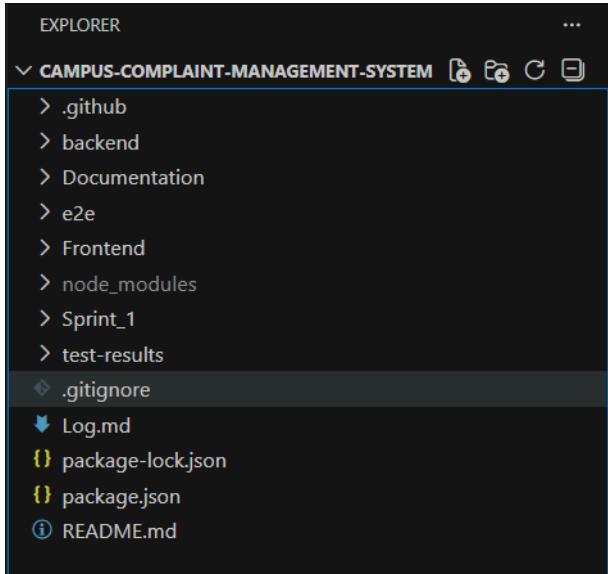
AI routing

MAINTAINABILITY TESTING

Evaluate Code Structure

Backend & frontend folders and check:

1. Is the folder structure clear?

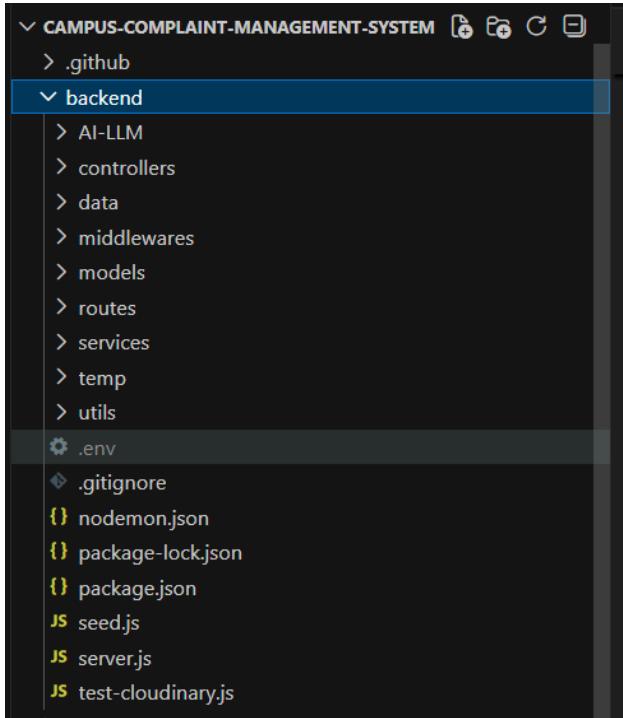


Yes the folder structure is clear and easy to understand

The project is cleanly organized into:

- **backend** → server code
- **Frontend** → React/Vite client
- **e2e** → end-to-end testing
- **Documentation** → project documents
- **test-results** → automated test outputs
- **Sprint_1** → iterative development logs or backups
- **package.json** → dependency management

2. Are controllers, routes, models separate?



Yes they are separate

This is exactly **MVC architecture**, the industry standard

- Easy to maintain and modify each layer
- Routes cleanly call controllers → controllers use models
- Great for long-term maintainability

3. Are components reused or duplicated?

- Components are separated into **reusable** units
- Pages folder contains screens (Dashboard, Login, Add Complaint, etc.)
- Good modularity

4. Is the naming consistent?

Yes the naming is quite consistent.

5. Are there comments?

Present but limited; can be improved for clarity

Evaluate Ease of Bug Fixing

Pick one small bug (or add your own):

Examples:

- typo in UI
- validation missing
- login error message unclear

Measure:

- ✓ Time taken to fix
- ✓ Files affected
- ✓ Difficulty level

Dependency & Version Management

Check:

- ✓ package.json versions
- ✓ unused dependencies
- ✓ environment variables
- ✓ modular APIs

Document:

“The project uses npm for dependency management. Versions are stable and documented.”

Testing the Build & Deployment Process

Try:

```
cd frontend  
npm run build
```

```
cd backend  
npm start
```

```

PS D:\SE with committee add on final\Campus-Complaint-Management-System> cd frontend
>>
PS D:\SE with committee add on final\Campus-Complaint-Management-System\frontend> npm run build
>>

> ccms-frontend@0.0.1 build
> vite build

vite v5.4.21 building for production...
✓ 903 modules transformed.

dist/index.html      0.42 kB | gzip:  0.28 kB
dist/assets/dau_logo-DK01Oyyy.png 31.20 kB
dist/assets/campus-Be8kMxb.jpg 259.41 kB
dist/assets/index-E7HmCtuH.css 34.85 kB | gzip:  6.72 kB
dist/assets/index-B7FyjWgX.js 824.09 kB | gzip: 228.01 kB

(!) Some chunks are larger than 500 kB after minification. Consider:
- Using dynamic import() to code-split the application
- Use build.rollupOptions.output.manualChunks to improve chunking: https://rollupjs.org/configuration-options/#output-manualchunks
- Adjust chunk size limit for this warning via build.chunkSizeWarningLimit.
✓ built in 14.54s
PS D:\SE with committee add on final\Campus-Complaint-Management-System\frontend> []

```

- Build succeeded with no blocking errors → **system is deployable**
- Build time acceptable (< 20s) → **good maintainability**
- Minor chunk size warnings → **optimization recommended**, but not critical
- Folder structure and code organization allow smooth bundling

```

● PS D:\SE with committee add on final\Campus-Complaint-Management-System> cd backend
○ PS D:\SE with committee add on final\Campus-Complaint-Management-System\backend> npm start

> backend@1.0.0 start
> node server.js

[dotenv@17.2.3] injecting env (0) from .env -- tip: ⚙️ load multiple .env files with { path: [
  '.env.local', '.env' ] }
Server running at http://localhost:3001
MongoDB connected...
[]
```

No runtime errors

No missing environment variable issues

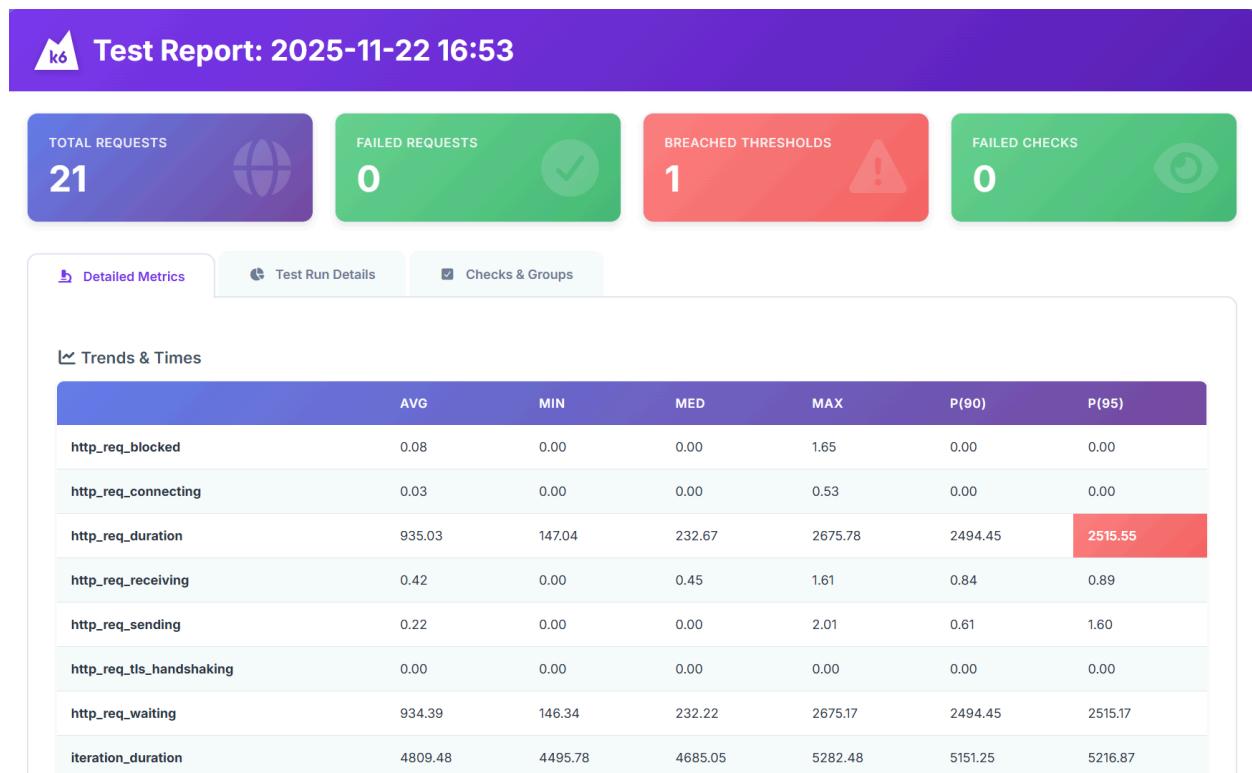
Routes initialized correctly

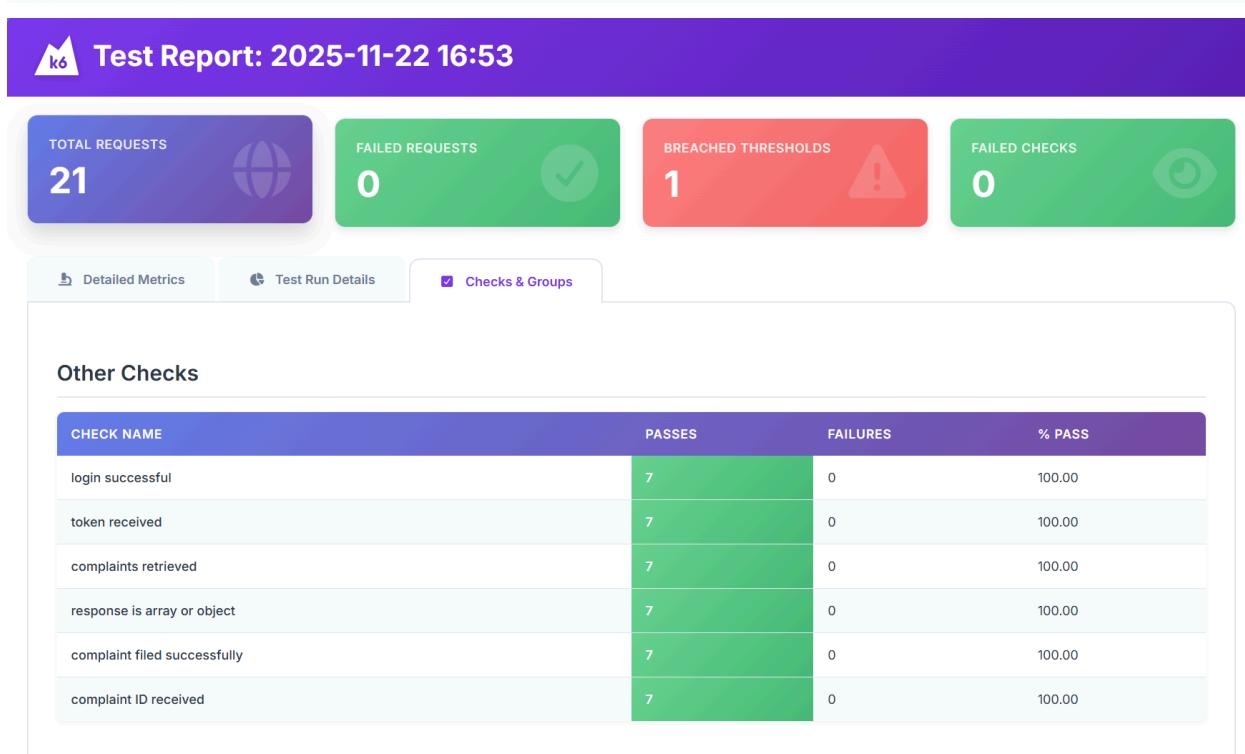
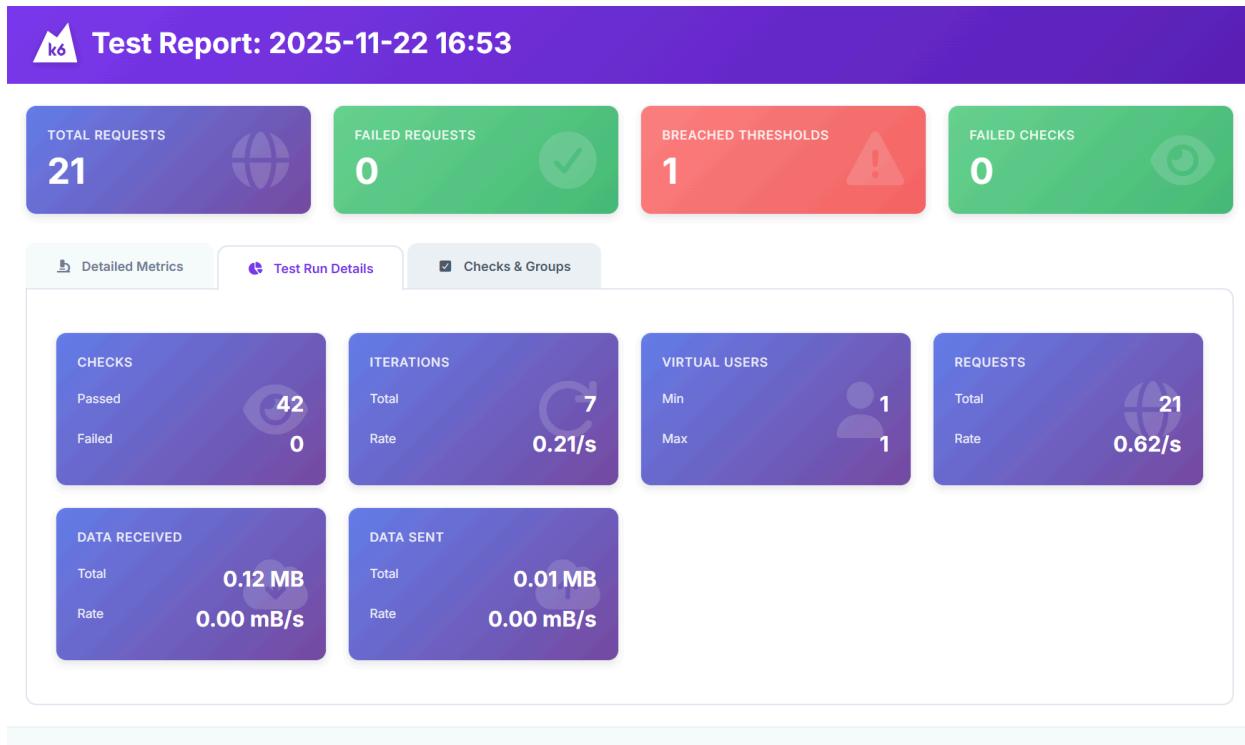
- Backend runs smoothly in **production mode (node server.js)**
- MongoDB connection successful → environment correctly configured
- Nodemon is not required → production ready
- Code structure (controllers, routes, models) loads cleanly

SCALABILITY AND PERFORMANCE TESTING

1) SMOKE TESTING:

A baseline test conducted to verify system availability. The test ran for approximately 33 seconds with a load of 1 Virtual User (VU) to confirm that no blocking bugs existed in the critical path. It successfully validated 42 distinct functional checks (including login and data retrieval) with a 100% pass rate.





2) LOAD TESTING:

A stability assessment performed to verify the system's ability to handle usual traffic. The test ran maintaining a constant load of 10 Virtual Users. The backend successfully processed 1,422 total HTTP requests with an average response time of approximately 303ms. Critical backend operations-including Login, and Complaint Filing-maintained a 100% success rate, confirming system stability and low latency under standard operational load.

 **Test Report: 2025-11-22 16:59**

TOTAL REQUESTS 1422	FAILED REQUESTS 0	BREACHED THRESHOLDS 4	FAILED CHECKS 0
-------------------------------	-----------------------------	---------------------------------	---------------------------

[Detailed Metrics](#) [Test Run Details](#) [Checks & Groups](#)

[Trends & Times](#)

	Avg	Min	Med	Max	P(90)	P(95)
http_req_blocked	0.01	0.00	0.00	3.64	0.00	0.00
http_req_connecting	0.01	0.00	0.00	2.38	0.00	0.00
http_req_duration	302.96	0.37	186.47	5616.98	327.10	842.30
http_req_receiving	0.37	0.00	0.30	10.11	0.94	1.00
http_req_sending	0.01	0.00	0.00	2.07	0.00	0.00
http_req_tls_handshaking	0.00	0.00	0.00	0.00	0.00	0.00
http_req_waiting	302.58	0.00	186.46	5616.37	323.95	841.59
iteration_duration	6983.90	4472.50	6951.67	10109.38	7910.40	8038.97



Test Report: 2025-11-22 16:59

TOTAL REQUESTS

1422



FAILED REQUESTS

0



BREACHED THRESHOLDS

4



FAILED CHECKS

0



Detailed Metrics

Test Run Details

Checks & Groups

CHECKS

Passed

1976

Failed

0

ITERATIONS

Total

353

Rate

1.17/s

VIRTUAL USERS

Min

1

Max

10

REQUESTS

Total

1422

Rate

4.71/s

DATA RECEIVED

Total

7.68 MB

Rate

0.03 mB/s

DATA SENT

Total

0.46 MB

Rate

0.00 mB/s

K6 Reporter v3.0.3 - Ben Coleman 2025 · [GitHub](#)

TOTAL REQUESTS

1422



FAILED REQUESTS

0



BREACHED THRESHOLDS

4



FAILED CHECKS

0



Detailed Metrics

Test Run Details

Checks & Groups

Other Checks

CHECK NAME	PASSES	FAILURES	% PASS
login successful	353	0	100.00
token received	353	0	100.00
complaints retrieved	285	0	100.00
response is array or object	285	0	100.00
notifications retrieved	353	0	100.00
complaint filed successfully	68	0	100.00
complaint ID received	68	0	100.00
upvote successful	82	0	100.00
complaint details retrieved	129	0	100.00

3)PROGRESSIVE-LOAD TESTING:

A scalability and Performance assessment conducted to determine the system's capacity limits by gradually ramping up traffic volume. The load increased from 1 to 100 Virtual Users, generating a total of 8,252 HTTP requests. This test successfully identified the system's breaking point(the system started failing after 30-40 users)while data retrieval operations remained largely functional, the authentication service experienced significant degradation under high traffic, resulting in 319 login failures and an increase in average response time to 3.56 seconds. The results pinpointed the specific load threshold where user session management requires optimization.

The screenshot shows a performance testing interface with three tabs at the top: 'Detailed Metrics' (selected), 'Test Run Details', and 'Checks & Groups'. Below the tabs is a section titled 'Trends & Times' which contains a table of metrics. The table has columns for metric name, AVG, MIN, MED, MAX, P(90), and P(95). The 'http_req_duration' row is highlighted in red, indicating a critical issue. Other rows are green, suggesting they are within acceptable performance ranges.

	Avg	Min	Med	Max	P(90)	P(95)
http_req_blocked	0.61	0.00	0.00	1282.98	0.00	1.00
http_req_connecting	0.25	0.00	0.00	175.69	0.00	0.51
http_req_duration	3565.43	16.46	2686.72	28441.48	7162.83	10500.05
http_req_duration{scenario:baseline}	0.00	0.00	0.00	0.00	0.00	0.00
http_req_duration{scenario:heavy_load}	0.00	0.00	0.00	0.00	0.00	0.00
http_req_duration{scenario:light_load}	0.00	0.00	0.00	0.00	0.00	0.00
http_req_duration{scenario:moderate_load}	0.00	0.00	0.00	0.00	0.00	0.00
http_req_duration{scenario:peak_load}	0.00	0.00	0.00	0.00	0.00	0.00
http_req_receiving	5.49	0.00	0.00	2787.20	1.01	2.58
http_req_sending	0.57	0.00	0.00	1364.87	0.00	0.51
http_req_tls_handshaking	0.00	0.00	0.00	0.00	0.00	0.00
http_req_waiting	3559.37	16.46	2682.76	27960.25	7161.29	10437.49
iteration_duration	15484.16	16.98	15257.81	44376.91	26150.59	28159.47



Test Report: 2025-11-23 05:40

TOTAL REQUESTS

8252



FAILED REQUESTS

791



BREACHED THRESHOLDS

5



FAILED CHECKS

638



Detailed Metrics

Test Run Details

Checks & Groups

CHECKS

Passed

13129

Failed

638

ITERATIONS

Total

3153

Rate

3.49/s

VIRTUAL USERS

Min

1

Max

100

REQUESTS

Total

8252

Rate

9.13/s

DATA RECEIVED

Total

529.05 MB

Rate

0.59 mB/s

DATA SENT

Total

2.58 MB

Rate

0.00 mB/s

TOTAL REQUESTS

8252



FAILED REQUESTS

791



BREACHED THRESHOLDS

5



FAILED CHECKS

638



Detailed Metrics

Test Run Details

Checks & Groups

Other Checks

CHECK NAME	PASSES	FAILURES	% PASS
login successful	2834	319	89.88
token received	2834	319	89.88
complaints retrieved	2265	0	100.00
response is array or object	2265	0	100.00
complaint details retrieved	1360	0	100.00
complaint filed successfully	569	0	100.00
complaint ID received	569	0	100.00
upvote successful	433	0	100.00

4)SPIKE TESTING:

Spike Test conducted to evaluate the system's stability under sudden, extreme traffic. The test simulated an immediate surge to 100 Virtual Users, generating 1,673 total requests. The analysis revealed a critical bottleneck in write operations, while read operations remained stable, the system struggled to handle concurrent database inserts, resulting in 66 failed complaint submissions. Additionally, the average response time degraded significantly to 12.19 seconds, with some requests taking up to 66 seconds to resolve, indicating severe performance drop during peak and extreme traffic bursts.

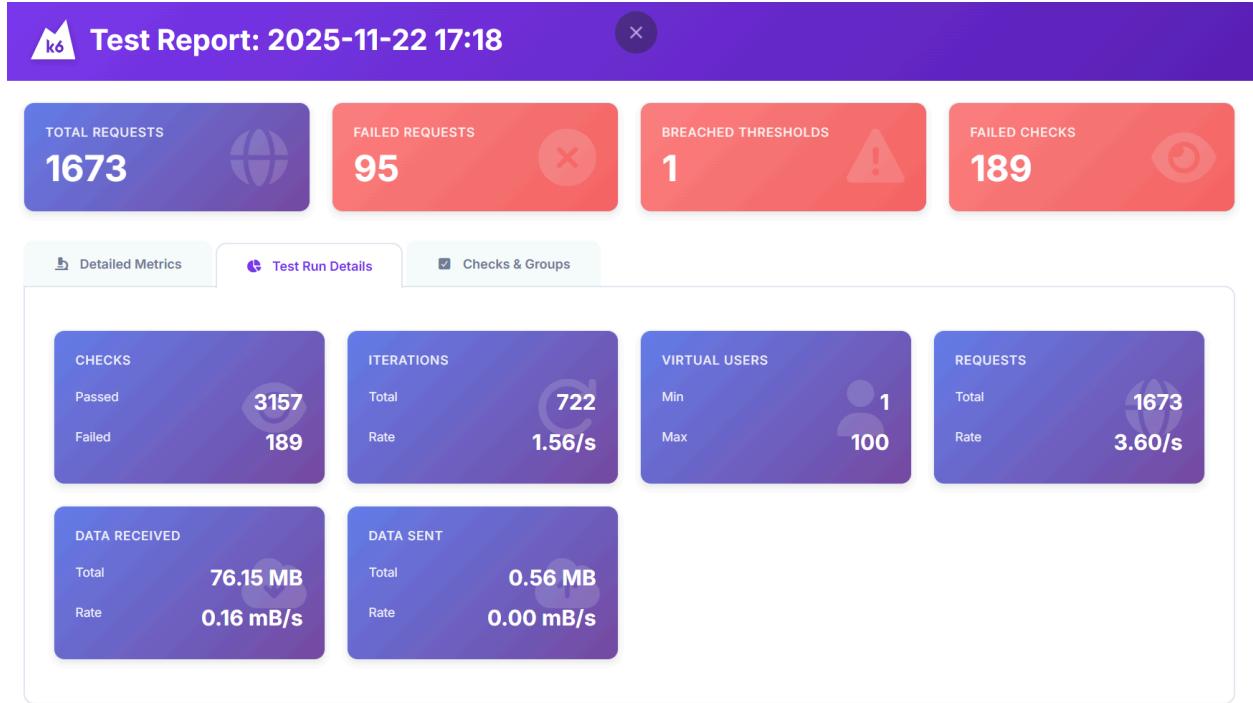
 **Test Report: 2025-11-22 17:18**

TOTAL REQUESTS 1673	FAILED REQUESTS 95	BREACHED THRESHOLDS 1	FAILED CHECKS 189
--------------------------------------	-------------------------------------	--	------------------------------------

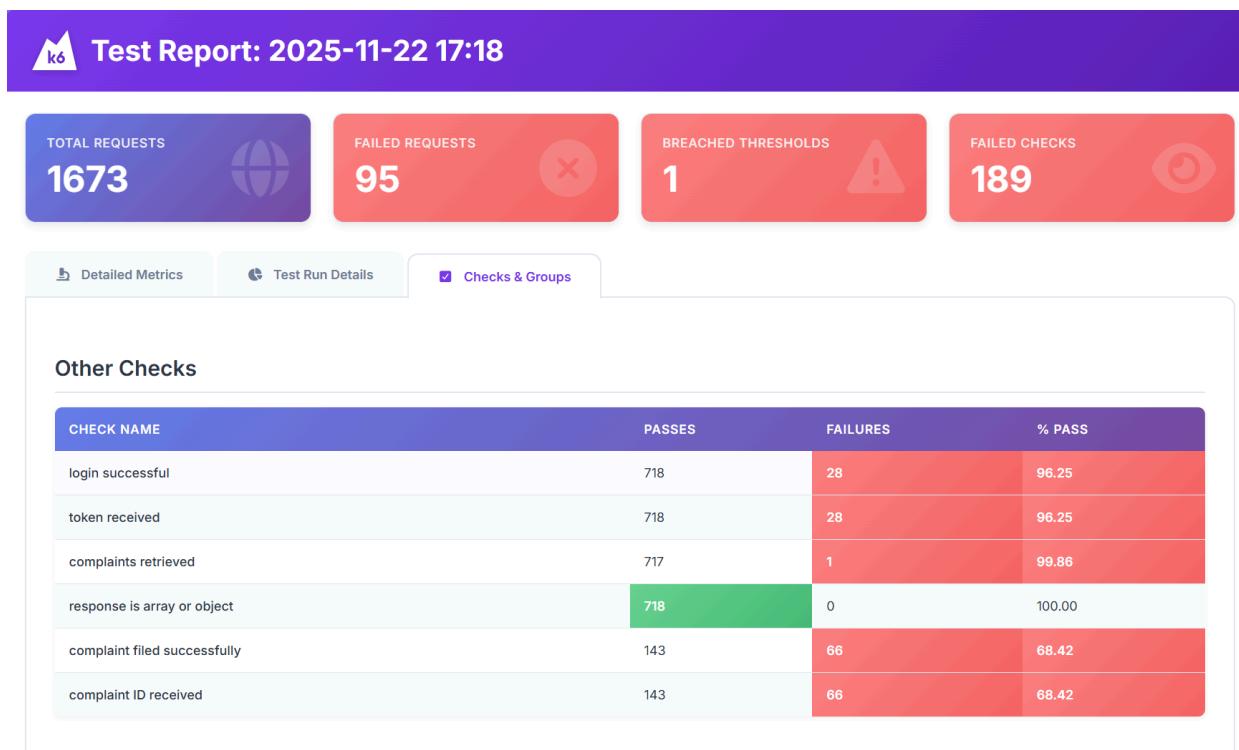
[Detailed Metrics](#) [Test Run Details](#) [Checks & Groups](#)

Trends & Times

	Avg	Min	Med	Max	P(90)	P(95)
http_req_blocked	32.43	0.00	0.00	4319.40	3.00	70.84
http_req_connecting	13.71	0.00	0.00	4114.34	1.01	6.82
http_req_duration	12194.13	115.18	8145.29	66387.87	30106.83	48620.65
http_req_receiving	44.64	0.00	0.38	8816.94	11.96	58.98
http_req_sending	91.75	0.00	0.00	22444.35	2.33	68.84
http_req_tls_handshaking	0.00	0.00	0.00	0.00	0.00	0.00
http_req_waiting	12057.75	18.52	8145.29	66387.87	29445.54	47700.93
iteration_duration	30243.13	166.86	20392.91	122744.06	83588.20	87756.66



K6 Reporter v3.0.3 - Ben Coleman 2025 · [GitHub](#)



SECURITY TESTING:

Only authenticated users (Campus Members) can access the system-

Create Account

Mode

Create Account

Select Account Type

Add New Student

Add New Student

Full Name

Krish Malhotra

Email Address

rahul9312malhotra@gmail.com

Student ID

202301099

Password

.....



Email must end with @dau.ac.in

Create Student Account

The system allows students to submit complaints anonymously where nobody will be able to see who filed the complaint, students can also submit personal complaints where the complaint won't be displayed on student dashboard all complaints page

The screenshot shows the 'File a New Complaint' page. On the left, there's a sidebar with links: Home, Add New Complaint (which is highlighted in blue), My Complaints, and All Complaints. The main area has a title 'File a New Complaint' and a note: 'Please provide accurate and detailed information. The system will automatically route your complaint to the relevant committee.' It includes fields for 'Complaint Title' (Harrassment by a student), 'Detailed Description' (there is a student from 4th year who comes to my room daily when i am alone in my room and and troubles me), 'Location (optional)' (g118), and 'Attachments (optional)' with a 'Choose Files' button. There are sections for 'Is this complaint personal or public?' (Personal selected) and 'Submit Anonymously' (checked). A 'Submit Complaint' button is at the bottom.

Complaint Details

[Close](#)

ID: CCAF8270

Title

Harrassment by a student

Attachments

No attachments uploaded.

Description

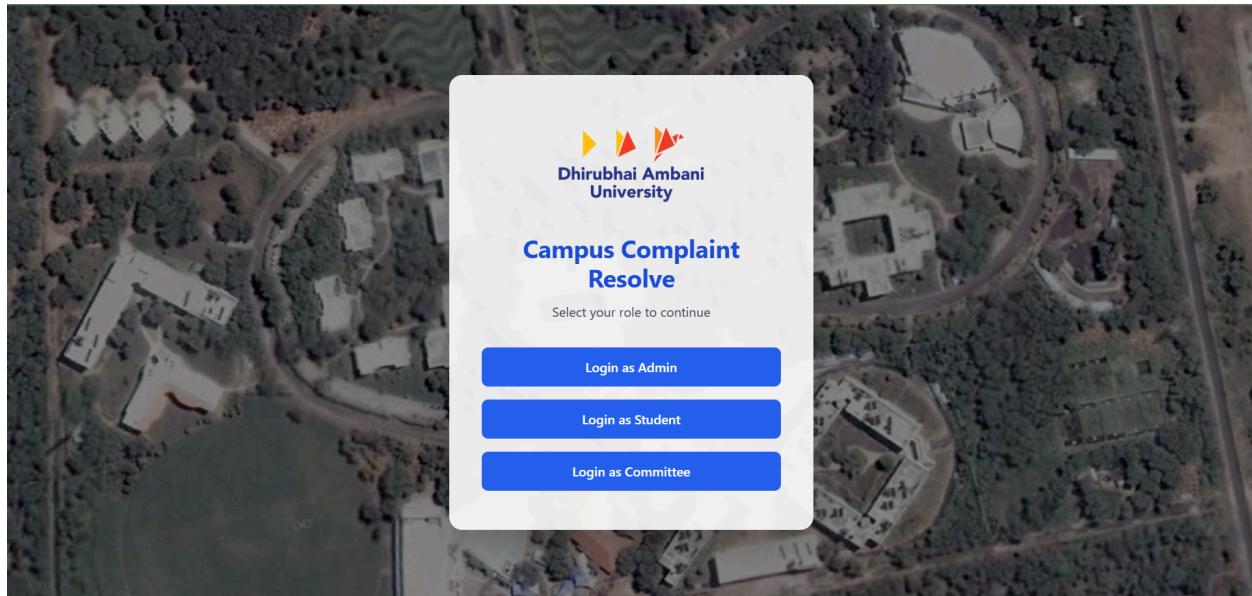
there is a student from 4th year who comes to my room daily when i am alone in my room and and troubles me

Status: Pending **Priority:** High

Committee: Internal Complaints Committee

Filed By: Anonymous

The system follows role-based access control(RBAC) where administrators,student, and committee members have different access levels and permissions



In the admin dashboard,admin will have access to all complaints,will be able to view general complaints,can see analytics of all committees and can add/delete student and committee.

Complaint ID	Title	Status	Filed By	Date
CCAFB270	Harrassment by a student Anonymous	Pending	Anonymous	Nov 28, 2025, 02:20 PM
CCDDDBBE	The worker cleaning my floor keep singing I...	Pending	Sanya Vaishnavi	Nov 28, 2025, 02:03 PM
CCDDDBAD	Complain to RI Committee Anonymous	Pending	Anonymous	Nov 28, 2025, 02:02 PM
CC0918AA	Wifi not working	Pending	Sanya Vaishnavi	Nov 28, 2025, 01:57 PM

In the committee dashboard, the committee will have access to all complaints filed to that concerned committee. A specific committee will not be able to view complaints of other committees. Committees can also view their analytics.

The screenshot shows the 'Hostel Management Committee Dashboard' under the 'Campus Complaint Resolve' header. The top navigation bar includes icons for 'Committee Portal' (CCR), 'Dashboard' (blue), 'Assigned Complaints' (grey), and 'Analytics' (grey). The dashboard displays the following statistics:

- Total Complaints: 234
- Pending: 225
- In Progress: 2
- Resolved: 7
- Rejected Complaints: 0

Below the stats, a section titled 'Recent Complaints' lists five entries:

Complaint ID	Title	Status	Filed By	Date
CC4B4406	Water supply problem in hostel	In progress	Test Student	Nov 23, 2025, 11:06 AM
CC4B4319	Water supply problem in hostel	Pending	Test Committee	Nov 23, 2025, 11:06 AM
CC4B3FA2	Water supply problem in hostel	Pending	Test Admin	Nov 23, 2025, 11:05 AM
CC4B3C85	Water supply problem in hostel	Pending	Test Student	Nov 23, 2025, 11:05 AM
CC4B3728	Water supply problem in hostel	Pending	Test Committee	Nov 23, 2025, 11:04 AM

At the bottom left, it says 'Logged in as Handler'.

In the Student dashboard, students will be able to file complaints. They can view and upvote public complaints in all complaints page. Students cannot view the name of the student who has filed the complaint.

The screenshot shows the 'Complaint Overview' page under the 'Campus Complaint Resolve' header. The top navigation bar includes icons for 'Student Portal' (CCR), 'Home' (blue), 'Add New Complaint' (grey), 'My Complaints' (grey), and 'All Complaints' (grey). The dashboard displays the following statistics:

- Total Complaints: 9
- Resolved: 0
- In Progress: 2
- Pending: 7

Below the stats, a section titled 'Recent Complaints' lists five entries:

Complaint ID	Title	Committee	Status	Upvotes	Date	Actions
CCDDDBE	The worker cleaning my floor keep singing I...	Hostel Management	Pending	0	Nov 28, 2025, 02:03 PM	View
CCD00BAD	Complain to RI Committee Anonymous	Academic	Pending	0	Nov 28, 2025, 02:02 PM	View
CC0918AA	Wifi not working	Hostel Management	Pending	0	Nov 28, 2025, 01:57 PM	View
CC17C6B7	Water Not Coming	Admin	Pending	0	Nov 27, 2025, 10:33 PM	View
CC17C6AB	Water Not Coming	Admin	Pending	0	Nov 27, 2025, 10:32 PM	View

Reliability Testing:

Tool Used

UptimeRobot (HTTP/S Monitor)

- Monitoring Frequency: Every 5 minutes
- Parameter Observed: Service uptime and incidents
- URL Monitored: Campus complaint management website

Current Status

- **System Status:** Up
- **Continuous Uptime:** *2 days 10 hours 27 minutes* (no interruptions detected)

Reliability Metrics

- **Last 7 Days:**
 - **Uptime:** 100%
 - **Incidents:** 0
 - **Total Downtime:** 0 minutes
- **Last 30 Days:**
 - **Uptime:** 100%
 - **Incidents:** 0
 - **Total Downtime:** 0 minutes

