

## Testing and Validation

### Student Names :

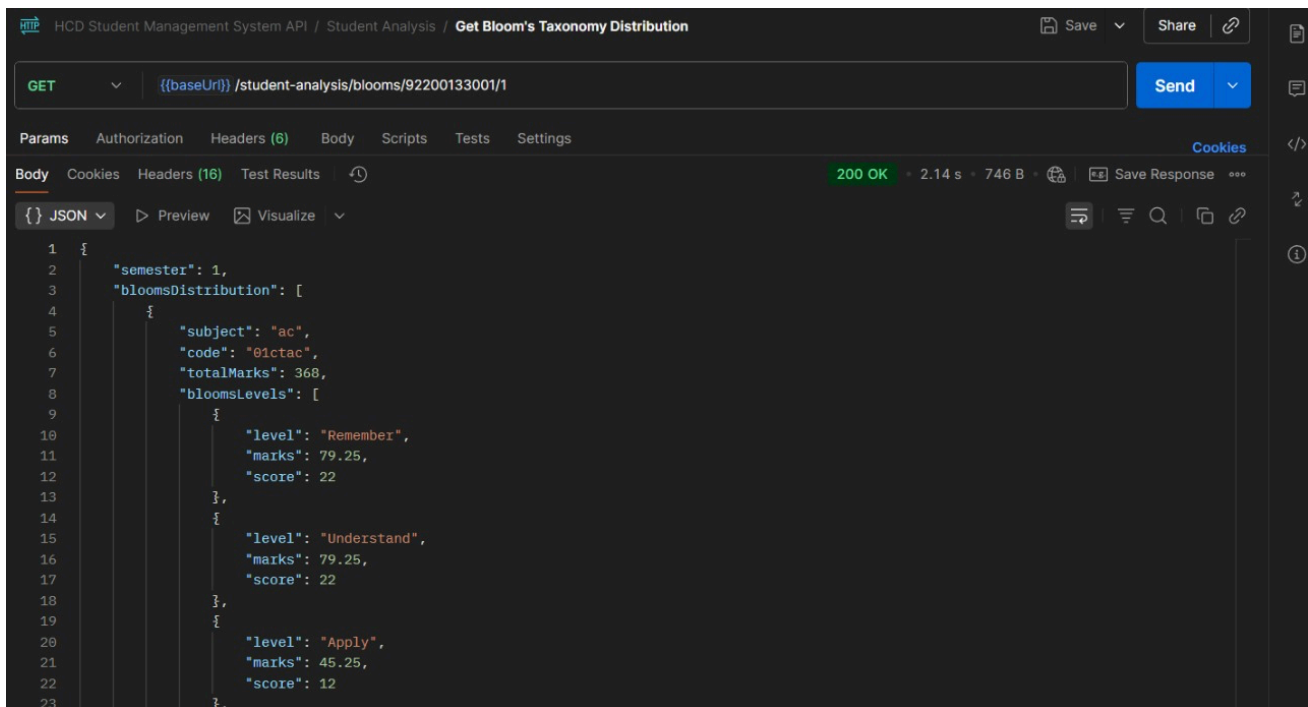
Harsh Doshi(92200133002)

Krish Mamtora (92200133022)

Rishit Rathod(92200133027)

### Unit testing

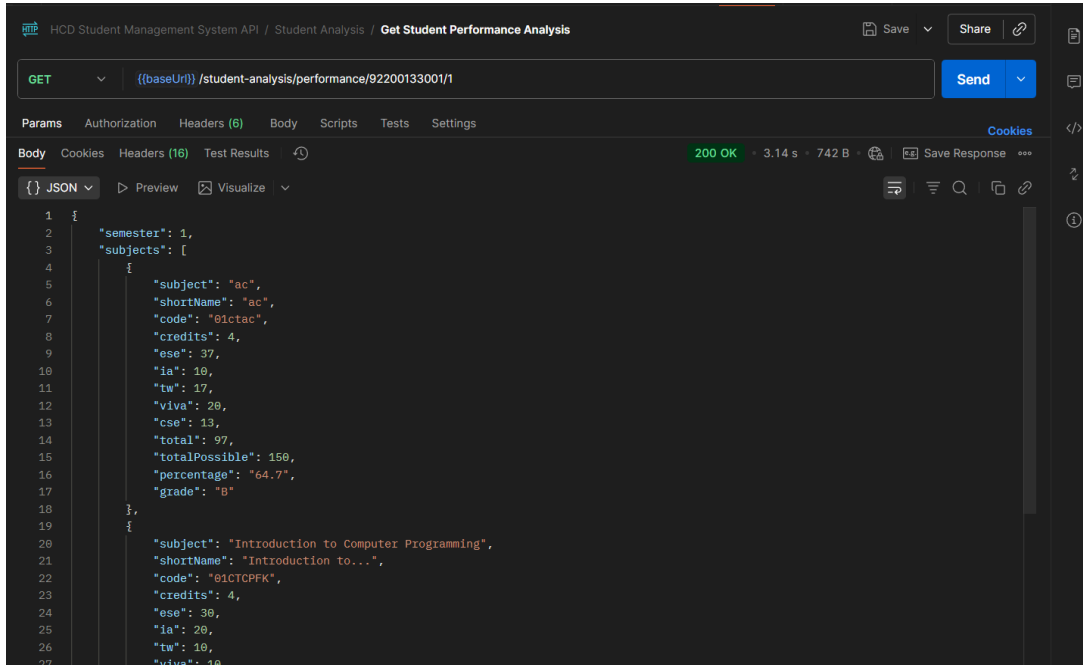
- 1) API testing to get blooms distribution marks and score data of student with response time 2.14 second and return 80 lines of records , Tools used : API postman



The screenshot shows a Postman interface for a GET request to the endpoint `{{baseUrl}}/student-analysis/blooms/92200133001/1`. The response status is **200 OK** with a response time of 2.14 s and a body size of 746 B. The response body is a JSON object:

```
1 {
2   "semester": 1,
3   "bloomsDistribution": [
4     {
5       "subject": "ac",
6       "code": "01ctac",
7       "totalMarks": 368,
8       "bloomsLevels": [
9         {
10          "level": "Remember",
11          "marks": 79.25,
12          "score": 22
13        },
14        {
15          "level": "Understand",
16          "marks": 79.25,
17          "score": 22
18        },
19        {
20          "level": "Apply",
21          "marks": 45.25,
22          "score": 12
23        }
24      ]
25    }
26  ]
27 }
```

- 2) API testing to get semester wise all subjects marks for each components with response time 3.14 second and return 40 lines of records, Tools used : API postman

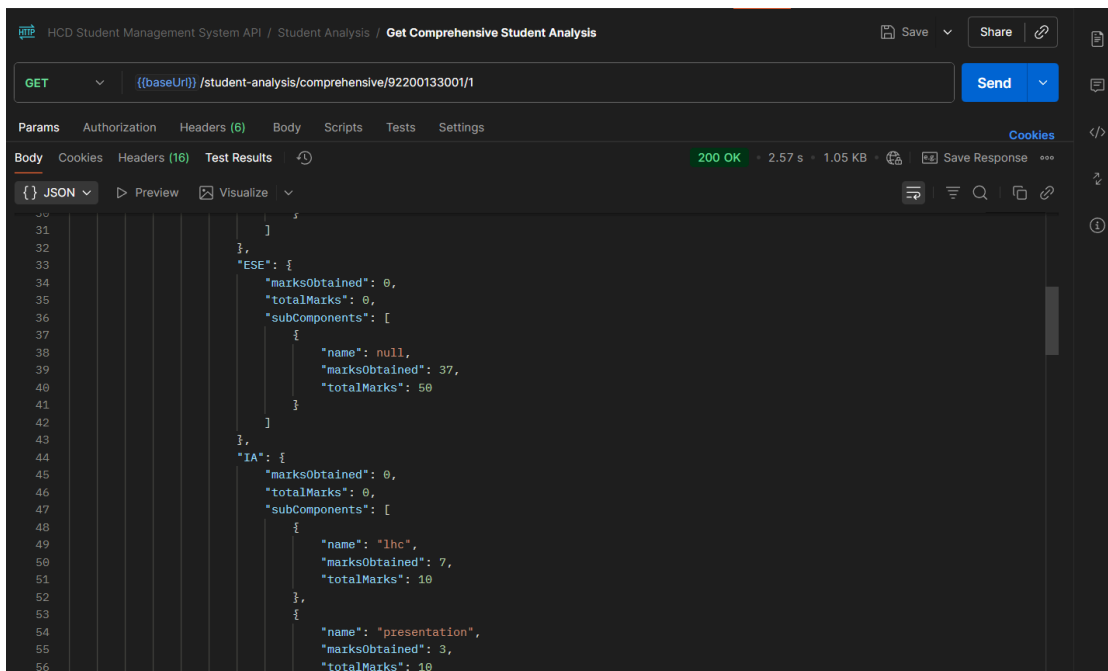


```

1 {
2   "semester": 1,
3   "subjects": [
4     {
5       "subject": "ac",
6       "shortName": "ac",
7       "code": "01ctac",
8       "credits": 4,
9       "ese": 37,
10      "ia": 10,
11      "tw": 17,
12      "viva": 20,
13      "cse": 13,
14      "total": 97,
15      "totalPossible": 150,
16      "percentage": "64.7",
17      "grade": "B"
18    },
19    {
20      "subject": "Introduction to Computer Programming",
21      "shortName": "Introduction to...",
22      "code": "01CTCPFK",
23      "credits": 4,
24      "ese": 30,
25      "ia": 20,
26      "tw": 10,
27      "viva": 10,

```

- 3) API testing to get marks obtained for student in particular semester for all components with response time 2.57s , Tools used : API postman

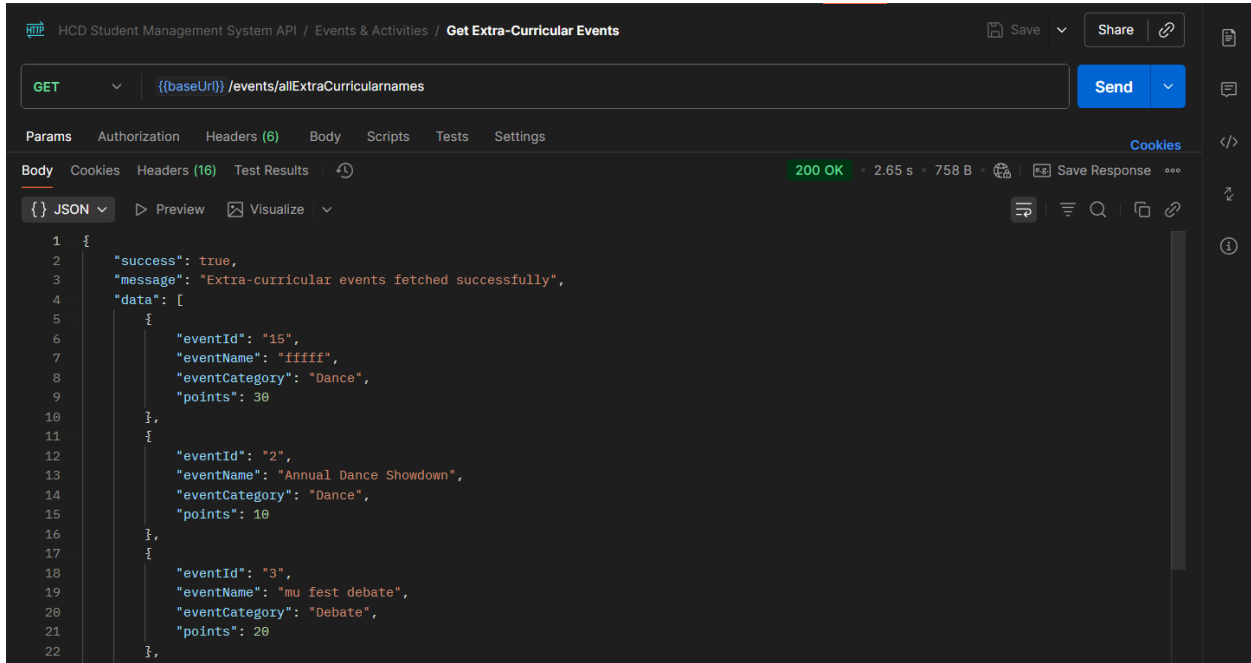


```

31   ],
32   },
33   "ESE": {
34     "marksObtained": 0,
35     "totalMarks": 0,
36     "subComponents": [
37       {
38         "name": null,
39         "marksObtained": 37,
40         "totalMarks": 50
41       }
42     ]
43   },
44   "IA": {
45     "marksObtained": 0,
46     "totalMarks": 0,
47     "subComponents": [
48       {
49         "name": "lhc",
50         "marksObtained": 7,
51         "totalMarks": 10
52       },
53       {
54         "name": "presentation",
55         "marksObtained": 3,
56         "totalMarks": 10

```

#### 4) API testing to get all event participations of any students along with its points



GET `{{baseUri}}/events/allExtraCurricularnames` **Send**

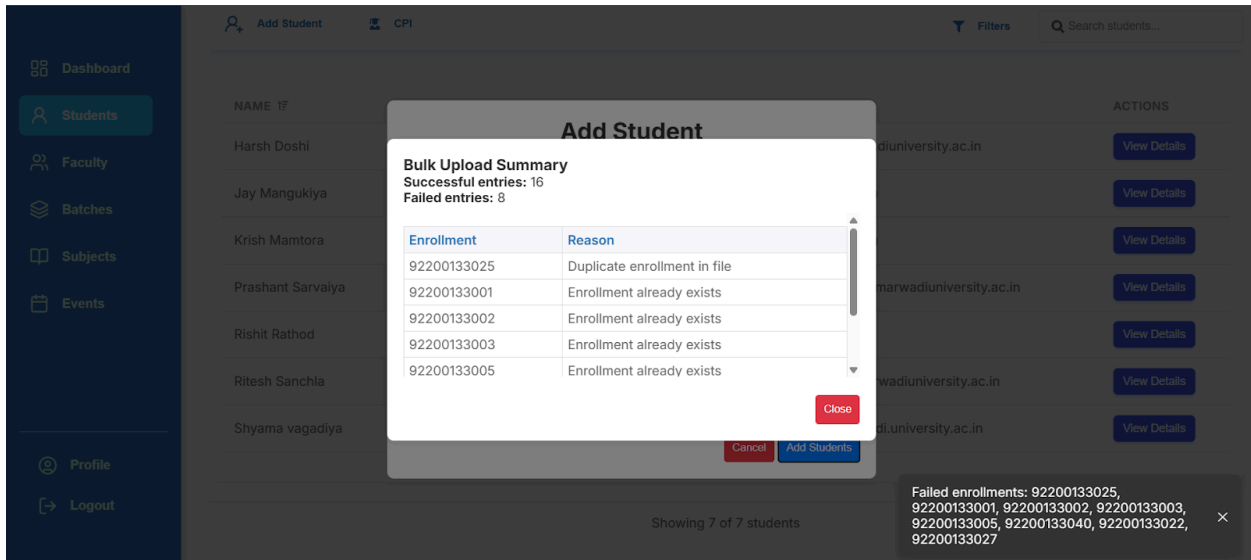
Params Authorization Headers (6) Body Scripts Tests Settings **JSON** Preview Visualize

200 OK • 2.65 s • 758 B • Save Response

```

1 {
2   "success": true,
3   "message": "Extra-curricular events fetched successfully",
4   "data": [
5     {
6       "eventId": "15",
7       "eventName": "fffff",
8       "eventCategory": "Dance",
9       "points": 30
10    },
11    {
12      "eventId": "2",
13      "eventName": "Annual Dance Showdown",
14      "eventCategory": "Dance",
15      "points": 10
16    },
17    {
18      "eventId": "3",
19      "eventName": "mu fest debate",
20      "eventCategory": "Debate",
21      "points": 20
22    }
23  ]
24 }
```

Testing for Student Upload feature : to warn user when duplicate students are entered with popup alert message ( notify successful and failed entries)



**Add Student**

**Bulk Upload Summary**  
Successful entries: 16  
Failed entries: 8

Enrollment	Reason
92200133025	Duplicate enrollment in file
92200133001	Enrollment already exists
92200133002	Enrollment already exists
92200133003	Enrollment already exists
92200133005	Enrollment already exists

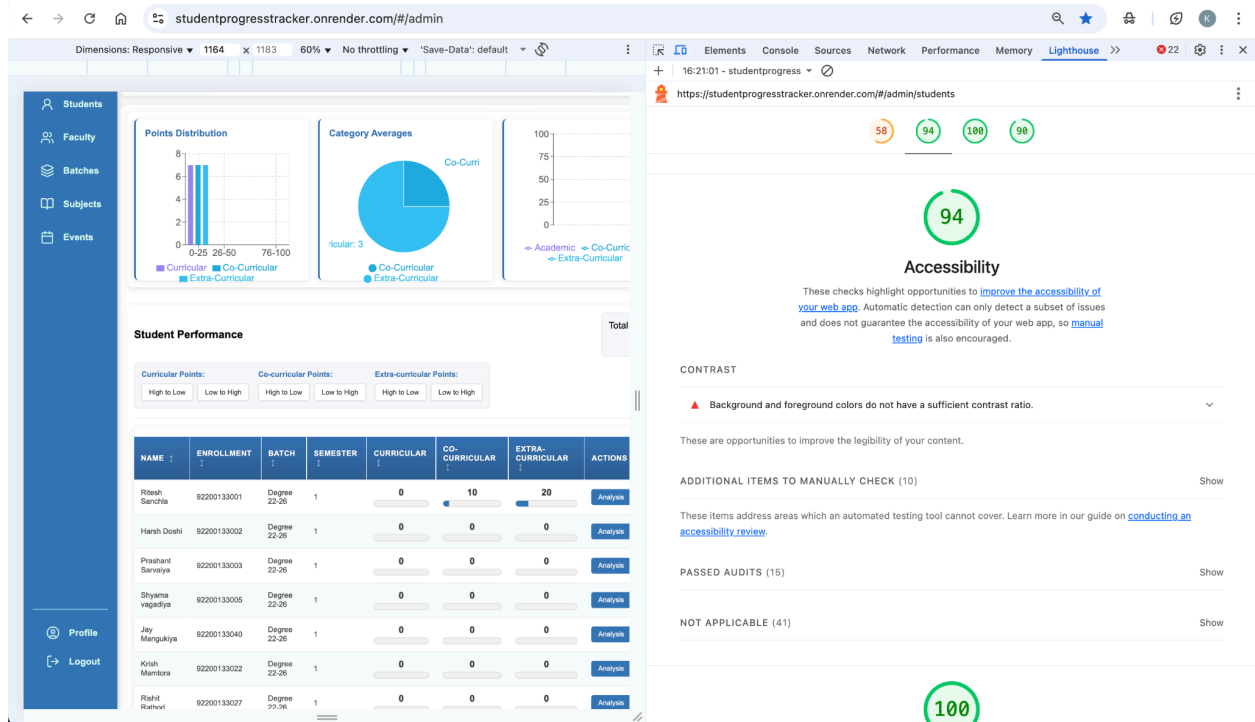
Failed enrollments: 92200133025, 92200133001, 92200133002, 92200133003, 92200133005, 92200133040, 92200133022, 92200133027

[illegible]

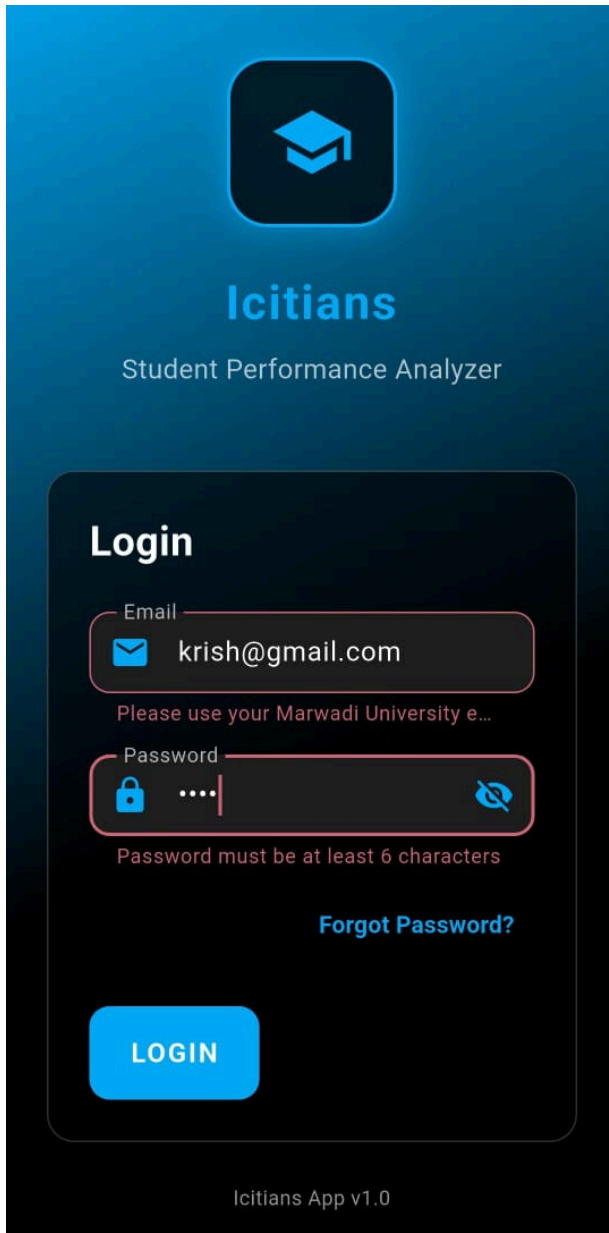
4

## Light house score

1. Best Practice : 100
2. Accessability : 94
3. SEO : 90
4. Performance : 58



In Mobile application validate user properly by ensuring user with marwadi email id can only enter and password should be minimum 6 character long.



**Icitions**  
Student Performance Analyzer

**Login**

Email

Please use your Marwadi University e...

Password

Password must be at least 6 characters

[Forgot Password?](#)

**LOGIN**

Icitions App v1.0