URLs		Responses	
Request	**Body**	**OK**	**Error**

POST api/login	username:______
	password:______
		Status = 200
		{
		‘username’: ______
		‘is_student’: (True/False)
		‘is_teacher’: (True/False)
		‘is_pending’:(True/False)
		‘registered’:(True/False)
		}
			Status = 400
			{
			‘A user with this username and password not exist’
			}

POST api/register	username:______
	password:______
	password2:______
	is_student:______
	is_teacher:______
		Status = 201
		{
		‘username’: ______
		‘is_student’: (True/False)
		‘is_teacher’: (True/False)
		‘is_pending’: (True/False)
		‘registered’:false
		}
			Status = 400
			{
			‘Validation Error’
			}

POST api/profile/student	Name:______
	Mobile:______
	Email: ______
	Year: ______
	Semester: ______
	Department: _____
	Enrollment: ______
	Section: ______
		Status = 201
		{
		‘username’: ______
		‘is_student’: (True/False)
		‘is_teacher’: (True/False)
		‘is_pending’: (True/False)
		}
			Status = 400
			{
			‘unable to add profile to user ’
			}

GET api/profile/student/<username>	--	Status = 200
		{
		‘Name’:______
		‘Mobile’:______
		‘Email’: ______
		‘Year’: ______
		‘Semester’: ______
		‘Department’: _____
		‘Enrollment’: ______
		‘Section’: ______
		}
			Status = 404
			{
			‘Profile not found’
			}

PUT api/profile/student	Name:______ Mobile:______ Email: ______ Year: ______ Semester: ______ Department: _____ Enrollment: ______ Section: ______	Status = 200	Status = 400

		{
		‘profile Successfully updated’
		}
			{
			‘<Error>’
			}

POST api/profile/teacher	Name:______ Mobile:______ Email: ______ Id:____	Status = 201	Status = 400

		{
		‘username’: ______
		‘is_student’: (True/False)
		‘is_teacher’: (True/False)
		‘is_pending’: (True/False)
		}
			{
			‘unable to add profile to user ’
			}

GET api/profile/teacher/<username>	--	Status = 200	Status = 404

		{
		‘Name’:______
		‘Mobile’:______
		‘Email’: ______
		‘Id’: ______
		}
			{
			‘Profile not found’
			}

PUT api/profile/teacher	Name:______ Mobile:______ Email: ______ Id: ______	Status = 200	Status = 400

		{
		‘profile Successfully updated’
		}
			{
			‘<Error>’
			}

GET api/attendance/student/<username>		Status = 200	Status = 400

		List of dictionary of following type which contains data based on current section
		{
		"Date": __,
		"Status": __,(A/P)
		"Enrollment":__,
		"Subject_Code": __,
		“Section": __,
		}
			[‘Some Error occur’]

Post api/attendance/student	enrollment:__, section:___	Status = 200	Status = 400

		List of dictionary of following type which contains data based on given section
		{
		"Date": __,
		"Status": __,(A/P)
		"Enrollment":__,
		"Subject_Code": __,
		“Section": __,
		}
			[‘Some Error occur’]

GET api/attendance/teacher/checkattendance/<str:section>/<str:subject_code>/<str:from_date>/<str:to_date>		Status = 200	Status = 500

		{
		‘<Subject_code>’:{
		‘<Enrollment>‘:{
		‘A’: <int>,
		‘P’: <int>,
		},
		……….
		}
		}
			<error>

GET Api/attendance/teacher/modifyattendance/ <str:date>/<str:section>/<str:subject_code>		Status : 200 [{ "Date": "2020-12-05", "Status": "A", "Enrollment": "S1", "Subject_Code": "1", "Section": "CSE-1A" }, ...]	Stauts : 500 **<Error>**
put api/attendance/teacher/modifyattendance	Date:2020-12-05 Status:A Enrollment:S3 Subject_Code:1 Section:CSE-1A	Status : 200 [“Data Updated”]	Stauts : 404 **<Error>**

Get
api/attendance/teacher/takeattendance/<str:section>/
<str:subject_code>/<str:class_id>
		Status : 200
		{
		"Desc.": "We'll call a function for marking attendance by face detection with the following parameter values",
		"Camera_id": "101",
		"Enrollment_List": [
		"S6",
		"S7",
		"S8",
		"S9",
		"S10"
		]
		}
		(Soon I’ll integrate the function with api)
			Stauts : 404
			<Error>

GET api/timetable/teacher/subject/<Department>/<Semester> <Department>: CSE,IT <Semester>: 1,2,3		**Status = 200** [{ "Subject_Code":"1", "Subject_Name":"Subject 1", "Semester": 1, "Department": "CSE" }, ………]	**Status:400** **<Error>**

I’ll add the Work flow Of project here as soon as possible….

For now I mentions some flow lines give an idea of how it actually works…

Flow Lines for Acute_Vision

User registeration will be in 2 steps first user created then profile added

User → login(post on api/login) →Home page (for student, teacher, teacher pending)

User → register(post on api/register)→ Profile Page (for student/teacher submit with post on api/profile/<usertype>)

User →(Home Page)→update profile(get on api/<usertype>/<username>)→Submit (PUT on api/profile/<usertype>)