

### Assignment3 WADL

```
export const loginController = async (req, res) => {
  try {
    const { emailID, password } = req.body;

    // Validation
    if (!emailID || !password) {
      return res.status(400).send({
        success: false,
        message: "Invalid email or password",
      });
    }

    // Check if the email is for a teacher
    const isTeacherEmail = emailID.includes("@pict.edu");

    // Check if user exists
    let foundUser;
    if (isTeacherEmail) {
      foundUser = await Teacher.findOne({ emailID });
    } else {
      foundUser = await Student.findOne({ emailID });
    }

    if (!foundUser) {
      return res.status(200).send({
        success: false,
        message: "Email is not registered. Please sign up",
      });
    }

    // Check if password matches
    const match = await comparePassword(password,
    foundUser.password);
    if (!match) {
      return res.status(200).send({
        success: false,
        message: "Invalid password. Please enter correct
password",
      });
    }

    // Send response
    res.status(200).send({
      success: true,
      message: "Login successful",
    });
  } catch (error) {
```

```

        console.log(error);
        res.status(500).send({
            success: false,
            message: "Error in login",
            error,
        });
    }
};

export const deleteAllStudentsController = async (req, res) => {
    console.log("Im Here!!!!");
    try {
        await Student.deleteMany({});

        res.status(200).json({
            success: true,
            message: 'All students deleted successfully',
        });
    } catch (error) {
        console.error(error);
        res.status(500).json({
            success: false,
            message: 'Error deleting all students',
            error: error.message,
        });
    }
};

import csv from 'csv-parser';
import { Readable } from 'stream';
import Student from '../model/student.model.js';

export const uploadStudentsController = async (req, res) => {
    try {
        const { file: studentFile } = req;

        console.log('Request file:', studentFile);

        if (!studentFile) {
            return res.status(400).send({
                success: false,
                message: 'File not found',
            });
        }

        const results = [];

```

```

    const stream =
Readable.from(studentFile.buffer.toString('utf-8'));

    stream
        .pipe(csv())
        .on('data', (data) => {
            console.log('Raw CSV data:', data);

            try {
                const subjects = JSON.parse(data['subjects']);
                const labs = JSON.parse(data['labs']);
                console.log('Parsed subjects:', subjects);
                console.log('Parsed labs:', labs);

                const newStudentData = {
                    rollNo: data['rollNo'],
                    name: data['name'],
                    attendance:
parseFloat(data['attendance']),
                    emailID: data['emailID'],
                    batch: data['batch'],
                    class: data['class'],
                    attendance: data['attendance'],
                    subjects,
                    labs
                };

                results.push(newStudentData);
            } catch (error) {
                console.error('Error parsing JSON:', error);
            }
        })
        .on('end', async () => {
            console.log('CSV reading completed. Contents:',
results);

            try {
                const insertedStudents = await
Student.insertMany(results, { maxTimeMS: 30000 });
                console.log('Data uploaded to MongoDB
successfully!', insertedStudents);

                res.status(201).send({
                    success: true,
                    message: 'Data uploaded to MongoDB
successfully!',
                    insertedStudents,
                });
            }

```

```

        } catch (error) {
            console.error('Error uploading data to
MongoDB:', error);
            res.status(500).send({
                success: false,
                message: 'Error uploading data to
MongoDB',
                error,
            });
        }
    });
} catch (error) {
    console.error(error);
    res.status(500).send({
        success: false,
        message: 'Error in student data upload',
        error,
    });
}
};

```

Output:

POST	http://localhost:8080/api/auth/login	Send	Status: 200 OK Size: 45 Bytes Time: 98 ms
Query	Headers 2	Auth	Body 1
JSON	XML	Text	Form
Form-encode	GraphQL	Binary	
JSON Content			Format
<pre> 1 { 2   "emailID" : "tanksalenishant5@pict.edu", 3   "password" : "securepassword123" 4 } </pre>			<pre> 1 { 2   "success": true, 3   "message": "Login successful" 4 } </pre>

GET	http://localhost:8080/api/student/33202	Send	Status: 200 OK Size: 862 Bytes Time: 16 ms
Query	Headers 2	Auth	Body 1
JSON	XML	Text	Form
Form-encode	GraphQL	Binary	
JSON Content			Format
<pre> 1 { 2   "emailID" : "tanksalenishant5@pict.edu", 3   "password" : "securepassword123" 4 } </pre>			<pre> 1 { 2   "success": true, 3   "message": "Student details retrieved successfully" 4   "data": { 5     "_id": "660c0aec4002bb7b4bec299", 6     "rollNo": "33202", 7     "name": "Adwait Desai", 8     "subjects": [ 9       { 10        "subjectName": "CC", 11        "UT1Marks": 27, 12        "UT2Marks": 28, 13        "attendance": 84, 14        "status": false 15      }, 16      { 17        "subjectName": "DSBDA", </pre>

DELETE	http://localhost:8080/api/student/deleteAllStudents	Send
Query	Headers 2	Auth
Query Parameters		
<input type="checkbox"/>	product	mnn
<input type="checkbox"/>	parameter	value

Status: 200 OK Size: 62 Bytes Time: 21 ms

Response

```
1 {
2   "success": true,
3   "message": "All students deleted successfully"
4 }
```

POST	http://localhost:8080/api/upload/students/upload-students	Send
Query	Headers 2	Auth
Body 1		
JSON	XML	Text
Form	Form-encode	GraphQL
Form Fields		
<input type="checkbox"/>	field name	value
<input checked="" type="checkbox"/>	studentFile	Choose file students101.csv
<input type="checkbox"/>	field name	Choose file Select file

Status: 201 Created Size: 3.5 KB Time: 71 ms

Response

```
1 {
2   "success": true,
3   "message": "Data uploaded to MongoDB successfully!"
4   "insertedStudents": [
5     {
6       "rollNo": "33201",
7       "name": "Adnan Patel",
8       "subjects": [
9         {
10          "subjectName": "CC",
11          "UT1Marks": 25,
12          "UT2Marks": 28,
13          "attendance": 67,
14          "status": false
15        },
16        {
17          "subjectName": "DSBDA",
```