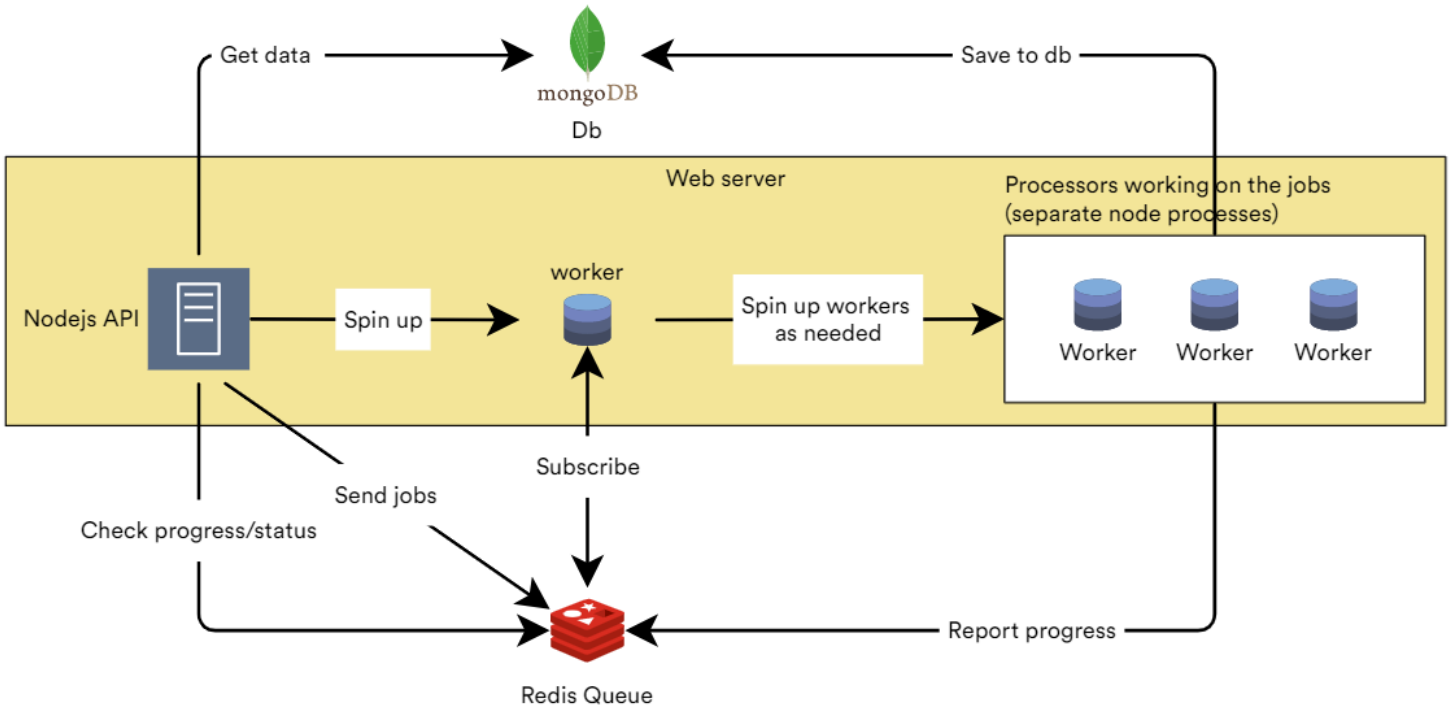
A- Architecture Diagram



B- Design description:

- To handle large file uploads the api should delegate the processing of these files to other processes, so we choose to use queue workers for this.

- The load can be handled by a single server for now, if we needed more we can spin up a new api server and add a load balancer or we can spin up these workers in a separate server altogether and with auto scaling (in AWS for example), these workers can be scaled out and in at any point anytime (such as peak periods)

Components
- API
- Redis queue (pub/sub)
- Mongodb (data storage)

Process steps (teacher):

1- API will run and create an internal worker process that will subscribe to Redis queue.

2- A teacher sends the request with file upload.

3- API will store the file (asynchronously) in a temp location and pass the request information to the queue.
4- API will be free to handle other requests.
5- The worker will receive the request from the queue and spin up a new internal process to handle the request.
6- If there are a lot of requests, the worker will spin up as much as needed (can be configured via concurrency) and will distribute the work in all processes in a round robin fashion, they will all work in parallel.

7- Apart from job processing, the worker process will also report progress back to the queue so it can be observed by the API.
8- Meanwhile the API can check the status and the progress
9- After the process finish processing it will save to the db.
10- Data will be available to students and teachers.

C- Code Assumptions

1- To keep things simple, I didn't create an admin user and auth, I assume that the teacher will be the admin, I created an endpoint to create the teacher
In the real world there will be an admin account to create the teachers and so on.

2- No need for the course's endpoint, the current prototype doesn't have an endpoint to manager courses (create, delete and so on), I store the course id with the grade for simplicity.

D- Design Estimation

1- Environment description:
- This is medium size college with roughly 10000 students in all grades.
- The number of courses is around 200 courses.
- Number of Teachers are roughly 140 teachers.
- Each course can take a round 50 students.
- Daily active students asking for grades are around 1000 students.
- Teachers probably will upload grades on the end of each semester and after quizzes and small exams.

2- Assumptions:
- Monthly active students are 15000 students.
- 50 % of students use it daily.
- Daily active teachers are 50 teachers.

- 10% of teachers upload grades every day.
- Data is stored for 5 years.

3- Estimation:
   - Daily active users = 50% * 15000 = 7500 query per second.
   - Grades upload = 10 % * 50 = 5 per second.
   - Peek (end of semester) = 2 * 5 = 15 per second.