**A) Client application registration requirement:** For any client application to be able to use/consume the rest api will require one time registration. JWT will be preferred approach.

**Acceptance criterion for the above requirement:**

* Any application which wishes to consume our api should do one time registration by furnishing some clientid, privatekey and obtain a token which should be used/presented for authentication of the api call.
* Should be able to create, list, edit or delete the tokens

**B) Functional requirements:**

1) **Main Function:** An async Endpoint or web method which will accept single/multiple files securely, which will be uploaded by the calling client (Can be Web / Application). Auto generate Task ID per file & returned the ID's for every file received in response.

**Acceptance criterion for the above functional requirement:**

* Should do JWT token based api authentication
* Should accept base64 encoding for file & data received or returned
* Should be able to upload multiple files simultaneously and saved concurrently in file folder present in the webroot directory.
* use efficient technique for file upload handling mechanism considering scalability and resources being consumed.
* Should generate & return file processing task id in response
* The generated task record should have data fields for Record sr no, task id, filename, received date-time stamp, task status, task result, completion date-time stamp, calling client-id, client IP-Address and any other fields deemed necessary
* Response time should be efficient

2) **Get** **Task Status Function:** An async Endpoint or web method which will return task status/progress queried by client application by providing input as single/multiple taskid(s) in an array or something

**Acceptance criterion for the above functional requirement:**

* Should do JWT token based api authentication
* Should accept and return data in base64 encoding
* Should read and return back values from task id, filename, task status, task result, result details, completion date-time stamp
* Response time should be efficient

3) **Get Task Logs Function:** An sync Endpoint or web method which will return logs of all file processing done for said period/date/month

**Acceptance criterion for the above functional requirement:**

* Should do JWT token based api authentication
* Should accept and return data in base64 encoding
* Should read and return back values from clientid, task id, filename, task status, task result, result details, receipt date-timestamp, completion date-time stamp
* Response time should be efficient

**Rest API Common criterions for acceptance:**

* Should support connection authentication using tokens
* Should allow asynchronous calls for performance and efficiency
* Should allow strict SSL (https) communication only with option to allow use or permitted/non vulnerable TLS version(1.2 or above) and non vulnerable Cipher suites
* Should do base64 encoding for any file/data being received or sent
* Should default log all api calls consumed/made by the clients for security audit purposes
* Should have all production grade logging, security, scalability and deployment considerations
* Project should be developed using IDE tools on Windows Desktop OS based platform
* Should have descriptions and comments in the source code for easy understanding and handover
* Should be developed using Java Springboot latest version with embedded Tomcat 10.x or higher & embedded no SQL for data
* no SQl database connection should be secured and should allow connection by other external modules from our product to update the task records
* Should checkin to our provided project or source control system

**Deliverables:**

* Should deliver qc tested production deployable springboot standalone system & application war file for windows server os based platform with setup steps or instructions
* Should deliver project solutions and source code with setup instructions
* Should deliver API documentation and usage details
* Should deliver documentation of high level design details and constructs used
* Should provide one time project handover to our team