Q1. How would you deploy this application in production?

A1. I would deploy this application to production using docker. These are the following reasons:

* **Efficient use of resources**: Containerized instance used way too less memory than virtual machines, start and stop quickly and they can be packed more densely on host hardware. This ensures less spending on infrastructure. The cost may depend on what applications are running and how resource intense it might be.
* **Faster delivery cycle**: Docker containers makes it easy to put new versions of software, with new features, into production quickly and to quickly roll back to previous versions.
* **Application portability**: Any host with Docker can run a docker container meeting all the requirements for running the application.

Q2. What other components would you want to add to make this production ready?

A2. Components to be considered to make this production ready:

* **Stability and Reliability**: To make sure your application does good in market, stability is an important aspect I would like to consider. Application being unstable/full of bugs would lead to negative reviews, resulting in downfall of application. Reliable is another important aspect as everyone looks for reliability of application.

For eg: Photo gallery application is reliable if the basic functionality i.e storing the photographs is done properly.

* **Scalability and Performance**: Before deploying an application, I would also consider laying out plan to make this application scalable in future. Performance of the application is also very important aspect. If your application slow/sluggish and doesn’t respond to input given then it might not be ideal to launch such application to production.
* **Fault Tolerance and Disaster Recovery**: Ensuring a fault tolerant system is crucial because the system might fail, and things might go sideways. Hence having a backup is important. To understand why the application failed we also need disaster recovery, what were the scenarios/factors leading to failure and have a solution ready if such failure occurs in future.
* **Monitoring**: We also need to monitor the application’s health once deployed.

Q3. How can this application scale with a growing dataset?

A3. To meet the increasing needs scaling out would be the preferred choice depending on the needs of the user or incoming requests.

Q4. How can PII be recovered later on?

A4. To recover the PII there is function written called encodeAndDecode. When *‘operation’* is passed as *‘decode’* it performs the decoding of encoded PII.

Q5. What are the assumptions you made?

A5. Assumed masking data means hiding the original data but still retaining the original value after decoding the data. To mask the data, I considered different algorithms like SHA256, MD5, base64 encoding. I chose base64 encoding because it is not complex compared to others and decoding the encoded part is also easy.