Internship Report

Submitted in partial fulfilment of the requirements for the award of degree of

Master of Computer Application

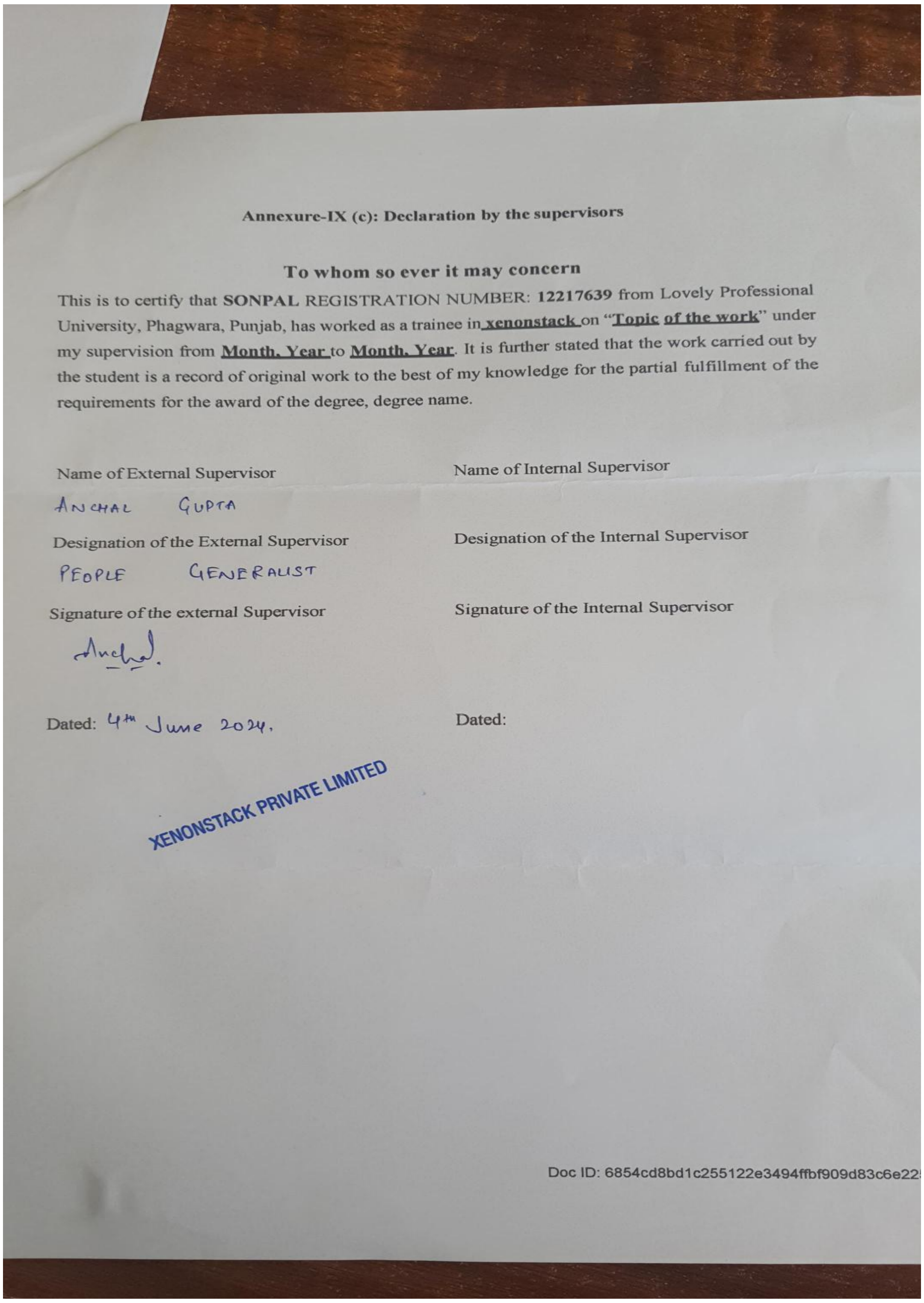
Submitted to:

LOVELY PROFESSIONAL UNIVERSITY PHAGWARA, PUNJAB

Submitted By:

SONPAL

Registration Number: 12217639



Annexure-IX (c): Declaration by the supervisors

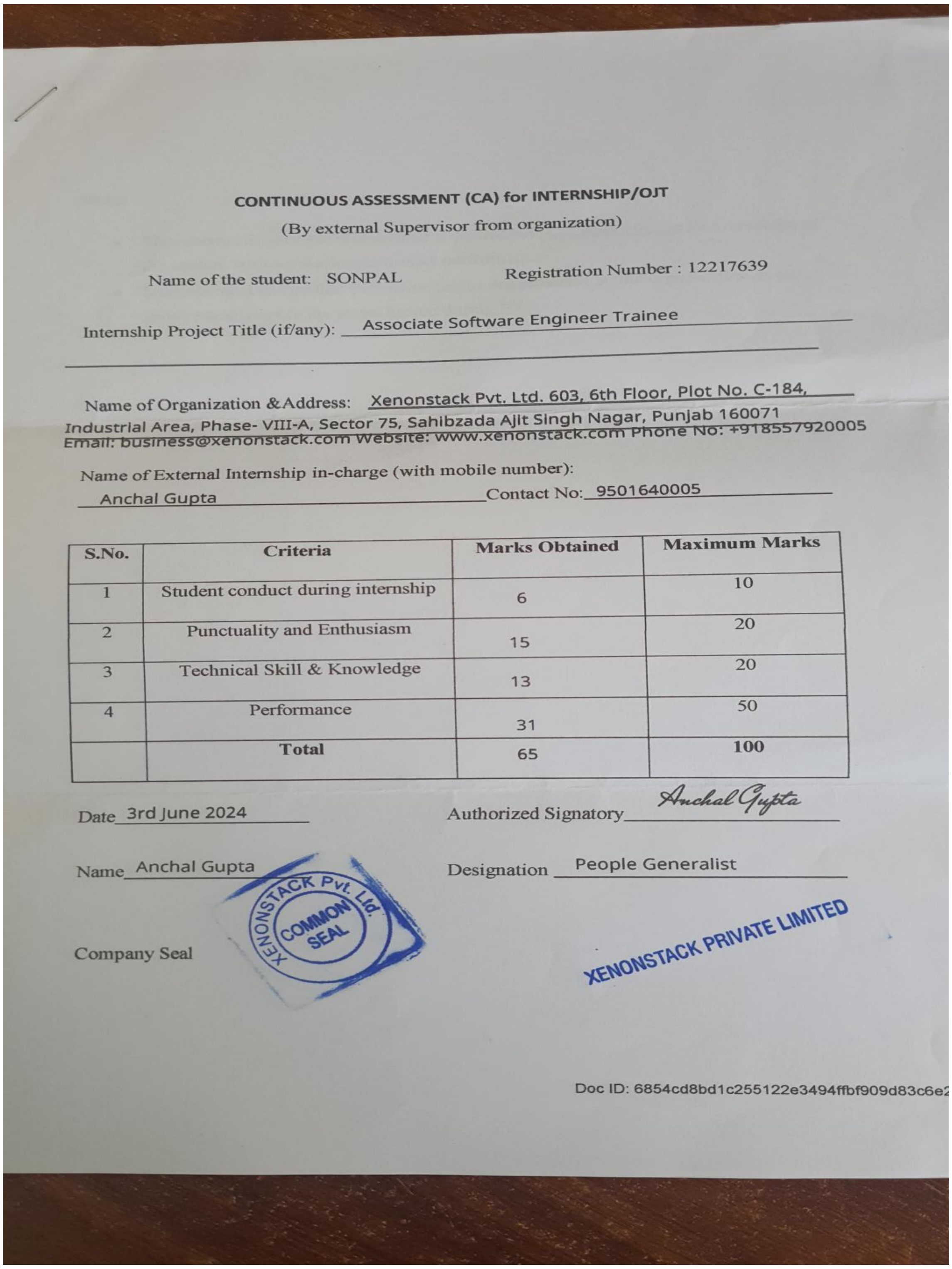
To whom so ever it may concern

This is to certify that **SONPAL** REGISTRATION NUMBER: **12217639** from Lovely Professional University, Phagwara, Punjab, has worked as a trainee in **xenonstack** on "**Topic of the work**" under my supervision from **Month, Year** to **Month, Year**. It is further stated that the work carried out by the student is a record of original work to the best of my knowledge for the partial fulfillment of the requirements for the award of the degree, degree name.

Name of External Supervisor	Name of Internal Supervisor
ANCHAL GUPTA	
Designation of the External Supervisor	Designation of the Internal Supervisor
PEOPLE GENERALIST	
Signature of the external Supervisor	Signature of the Internal Supervisor
Anchal.	
Dated: 4th June 2024.	Dated:

XENONSTACK PRIVATE LIMITED

Doc ID: 6854cd8bd1c255122e3494ffbf909d83c6e22



CONTINUOUS ASSESSMENT (CA) for INTERNSHIP/OJT

(By external Supervisor from organization)

Name of the student: SONPAL   Registration Number : 12217639

Internship Project Title (if/any): Associate Software Engineer Trainee

Name of Organization &Address: Xenonstack Pvt. Ltd. 603, 6th Floor, Plot No. C-184, Industrial Area, Phase- VIII-A, Sector 75, Sahibzada Ajit Singh Nagar, Punjab 160071 Email: business@xenonstack.com Website: www.xenonstack.com Phone No: +918557920005

Name of External Internship in-charge (with mobile number):
Anchal Gupta   Contact No: 9501640005

S.No.	Criteria	Marks Obtained	Maximum Marks
1	Student conduct during internship	6	10
2	Punctuality and Enthusiasm	15	20
3	Technical Skill & Knowledge	13	20
4	Performance	31	50
	Total	65	**100**

Date 3rd June 2024   Authorized Signatory Anchal Gupta

Name Anchal Gupta   Designation People Generalist

Company Seal   XENONSTACK Pvt. Ltd. COMMON SEAL   XENONSTACK PRIVATE LIMITED

Doc ID: 6854cd8bd1c255122e3494ffbf909d83c6e2

Table of Contents

ACKNOWLEDGEMENT

I extend my heartfelt gratitude to Xenonstack Pvt. Ltd. for providing me with the platform to apply my academic knowledge in a real-world setting. The innovative projects and supportive atmosphere have really enhanced my educational experience.
I express my sincere appreciation to my mentors, Spandan Saha, Kuldeep Kaur, Nitin Agarwal and Satyaki Chakrabarty, whose guidance and expertise have been indispensable throughout my internship. Their mentorship not only provided me with direction but also instilled in me a deeper understanding of cloud security principles.

I express my gratitude to my fellow members of the Internship team for their steadfast assistance and cooperation. Their openness to collaborate and exchange knowledge has really aided in my own progress. I am appreciative of Xenonstack Pvt. Ltd.'s tools and assistance, which allowed me to take on various challenging technical projects and obtain practical experience. Having access to cutting edge technologies and educational resources has been priceless.

Reflecting on my internship journey, I am amazed by the depth of knowledge I have acquired and the skills I have honed. Beyond technical expertise, this experience has taught me the importance of adaptability, collaboration, and continuous learning in a dynamic industry like cloud security.

Finally, I would like to express my sincere gratitude to my mentor of Xenonstack as well as my colleagues for their constant support and direction during this internship. I have no doubt that the knowledge and expertise I've received from this internship will be a great asset to me in the future.

SONPAL

12217639

ABSTRACT

This internship report offers a comprehensive overview of my experience at Xenonstack, where I worked as an ASE trainee Intern within the Internship team. The internship centered on monitoring and optimizing technical performance in various technology, a critical aspect of data integration in organizations. It begins by introducing Linux and cloud technology, setting the stage for the internship objectives: gaining practical experience in monitoring linux environments, identifying performance bottlenecks, and implementing optimization strategies.
Throughout the internship, various methodologies were employed, including the use of monitoring tools, performance analysis techniques, and performing in various technology. Challenges encountered, such as troubleshooting issues and fine-tuning configurations, were met with strategic approaches to overcome them. The outcomes included improved performance metrics, enhanced system stability, and valuable insights into various technology like cloud, data engineering, software engineering, react js and software testing.

Additionally, the report discusses my final semester internship experience, emphasizing hands-on various technology like cloud, data engineering, software engineering, react js and software testing and collaboration with industry professionals at XS. Projects focused on enhancing data security, utilizing analysis tools and technologies, and participating in team discussions and brainstorming sessions.
The implications of both internship experiences extend beyond academic learning, contributing to personal and professional growth. Practical exposure has equipped me with valuable skills and insights relevant to the evolving landscape of data management and security. The report concludes with reflections on the significance of these experiences for future career aspirations.

Chapter 1: Company Details and Role Details

XenonStack is all about using the latest tech to help businesses grow. They're experts in things like artificial intelligence, cloud computing, and cybersecurity. What's cool is that they work closely with each client to understand exactly what they need, whether it's streamlining processes or beefing up data security. They've got a solid track record, having worked with big names in healthcare, finance, and more.

What sets us apart is our customer-centric approach. We collaborate closely with our clients to understand their unique challenges and goals, and then tailor our solutions to meet their specific needs. Whether it's optimizing processes, enhancing data analytics capabilities, or building scalable and secure applications, XenonStack is dedicated to helping businesses achieve digital excellence.

1.1 Principles & Values

The leadership team of XS is passionate about providing an ecosystem that embraces a continuous experimentation approach that empowers the growth of the community and region. Our diverse environment helps organizations build Agile and Scalable platforms that leverage industry-leading best practices.

The industry works on various industry technology in which they build some of the big projects in the following industry:

- **Digital Retail Services**
- **Digital Manufacturing Services**
- **AI solutions in healthcare**
- **Technology Solutions for Travel and Hospitality Industry**
- **Digital Transformation in Banking and Financial Services**

Chapter 2: Project Description

During our internship, we have worked on several technologies that are constantly being used in the modern tech world, such as:

1. Library Management System – Backend

Being an online library system, Platform gives the capability for anyone to create their own library. For each Library, we would have 2 type of user profiles (LibraryAdmin, Readers). Where Owner or Creator of the Library can directly onboard LibraryAdmin and he/she can share a sign-up portal to Readers. Here are some key points about our LIbrary Management System:

1. **Purpose**: To design and build an online library management system for everyone to create their own library and onBoard admins to manage the library.
2. **Features**:

    - User Profiles:
        - **LibraryAdmin:** As the owner or creator of a library, the LibraryAdmin has full control over library management tasks such as adding books, managing users, and configuring settings.
        - **Readers:** Readers have access to the library content shared by the LibraryAdmin. They can browse through the collection, borrow books, and interact with other users through discussion forums.
    - Easy Onboarding:
        - The system provides a simple and intuitive sign-up process for LibraryAdmins, allowing them to quickly set up their libraries and start sharing content.

        - LibraryAdmins can effortlessly invite Readers to join their libraries by sharing a unique sign-up portal, making the onboarding experience seamless and user-friendly.

    - Comprehensive Book Catalog:
        - The platform hosts a vast collection of books from various publications and authors, including different editions and versions.
        - Each book is assigned a standard ISBN number, ensuring global uniqueness and facilitating easy identification within the system.

3. **Technologies Used**
    - Backend – Go/GoLang:

        - Go, with its simplicity, performance, and concurrency features, serves as the backbone of the system's backend architecture. It enables efficient handling of concurrent requests and seamless integration with other components.

    - Frontend – React Js:

        - React.js powers the dynamic and interactive user interface of the Online Library Management System. Its component-based architecture and virtual DOM make it ideal for building responsive and scalable web applications.

- Database – Postgres:

    - A robust database system is employed to store and manage the extensive collection of books, user profiles, and other relevant data. Postgres provides the necessary reliability, scalability, and performance for handling large volumes of information.

Fig 1.1 – Go and Gin Framework

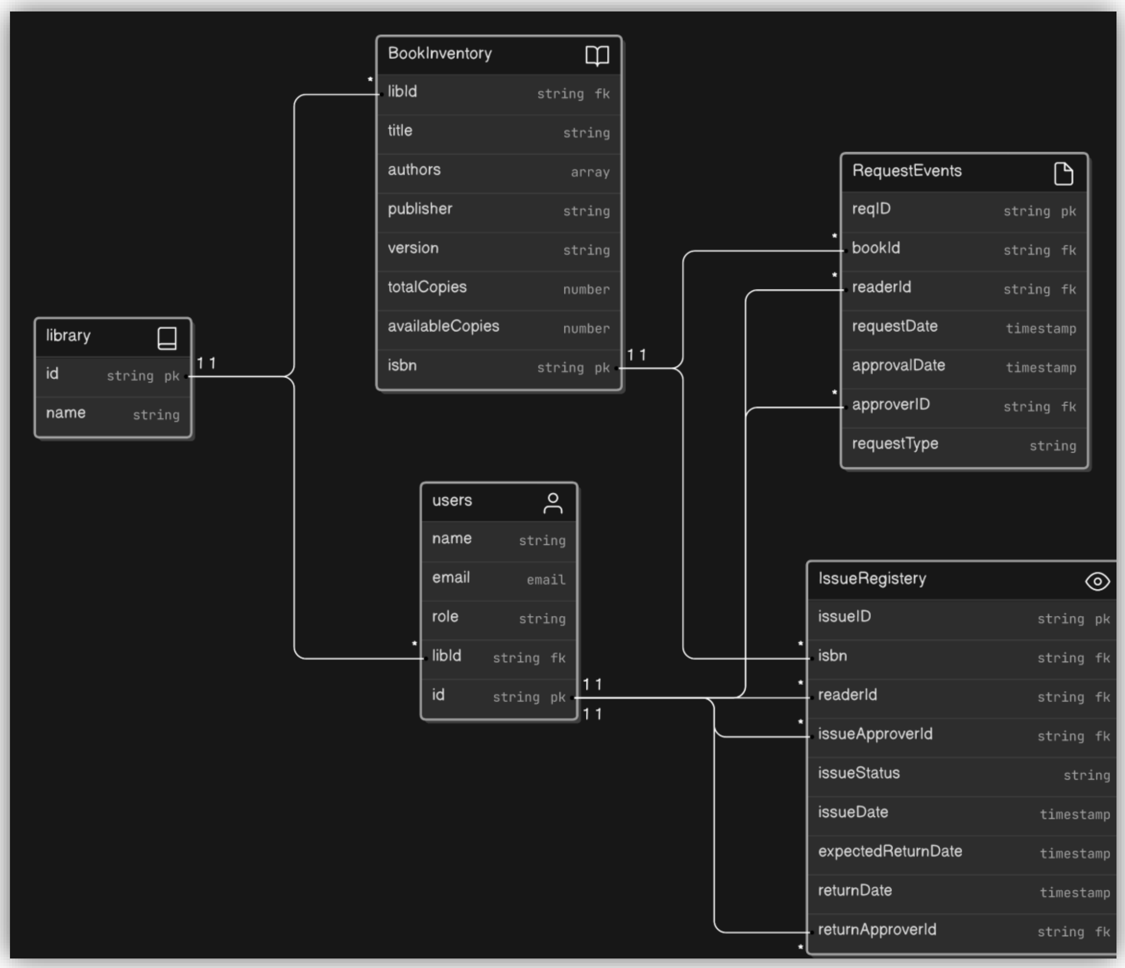
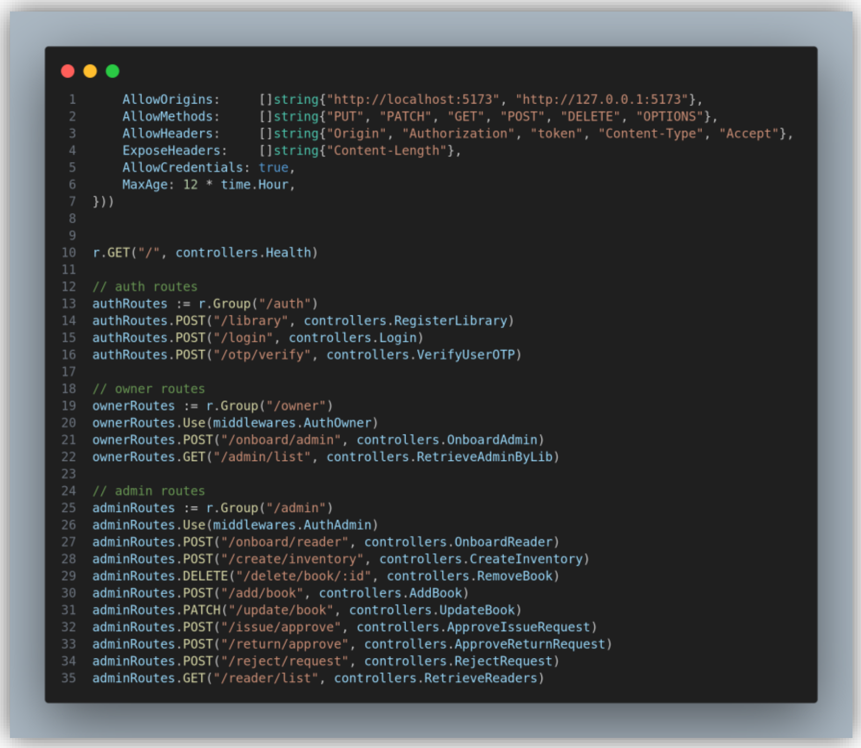
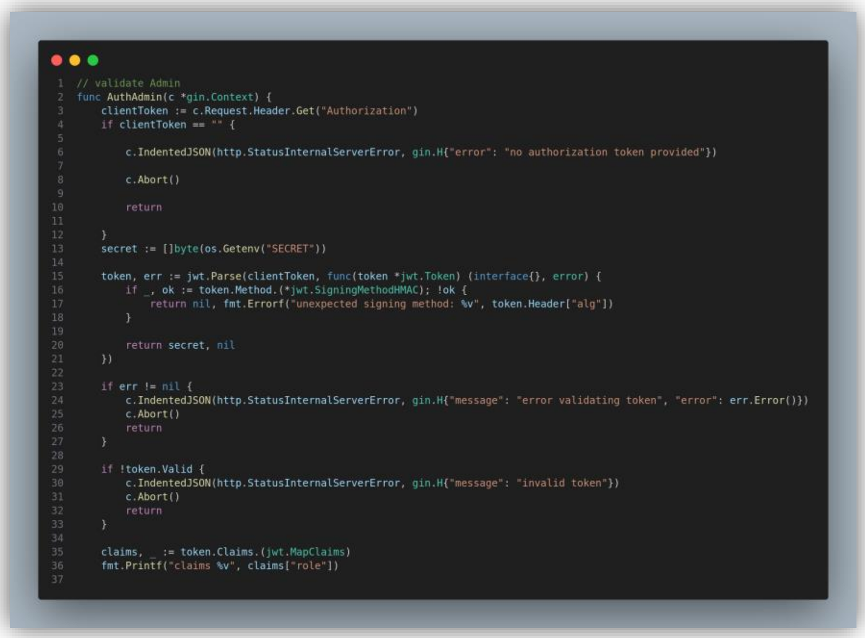


Fig 1.2 - ER Diagram of the Project



```go
    AllowOrigins:     []string{"http://localhost:5173", "http://127.0.0.1:5173"},
    AllowMethods:     []string{"PUT", "PATCH", "GET", "POST", "DELETE", "OPTIONS"},
    AllowHeaders:     []string{"Origin", "Authorization", "token", "Content-Type", "Accept"},
    ExposeHeaders:    []string{"Content-Length"},
    AllowCredentials: true,
    MaxAge: 12 * time.Hour,
}))


r.GET("/", controllers.Health)

// auth routes
authRoutes := r.Group("/auth")
authRoutes.POST("/library", controllers.RegisterLibrary)
authRoutes.POST("/login", controllers.Login)
authRoutes.POST("/otp/verify", controllers.VerifyUserOTP)

// owner routes
ownerRoutes := r.Group("/owner")
ownerRoutes.Use(middlewares.AuthOwner)
ownerRoutes.POST("/onboard/admin", controllers.OnboardAdmin)
ownerRoutes.GET("/admin/list", controllers.RetrieveAdminByLib)

// admin routes
adminRoutes := r.Group("/admin")
adminRoutes.Use(middlewares.AuthAdmin)
adminRoutes.POST("/onboard/reader", controllers.OnboardReader)
adminRoutes.POST("/create/inventory", controllers.CreateInventory)
adminRoutes.DELETE("/delete/book/:id", controllers.RemoveBook)
adminRoutes.POST("/add/book", controllers.AddBook)
adminRoutes.PATCH("/update/book", controllers.UpdateBook)
adminRoutes.POST("/issue/approve", controllers.ApproveIssueRequest)
adminRoutes.POST("/return/approve", controllers.ApproveReturnRequest)
adminRoutes.POST("/reject/request", controllers.RejectRequest)
adminRoutes.GET("/reader/list", controllers.RetrieveReaders)
```

Fig 1.3 - Main.go File Snippet



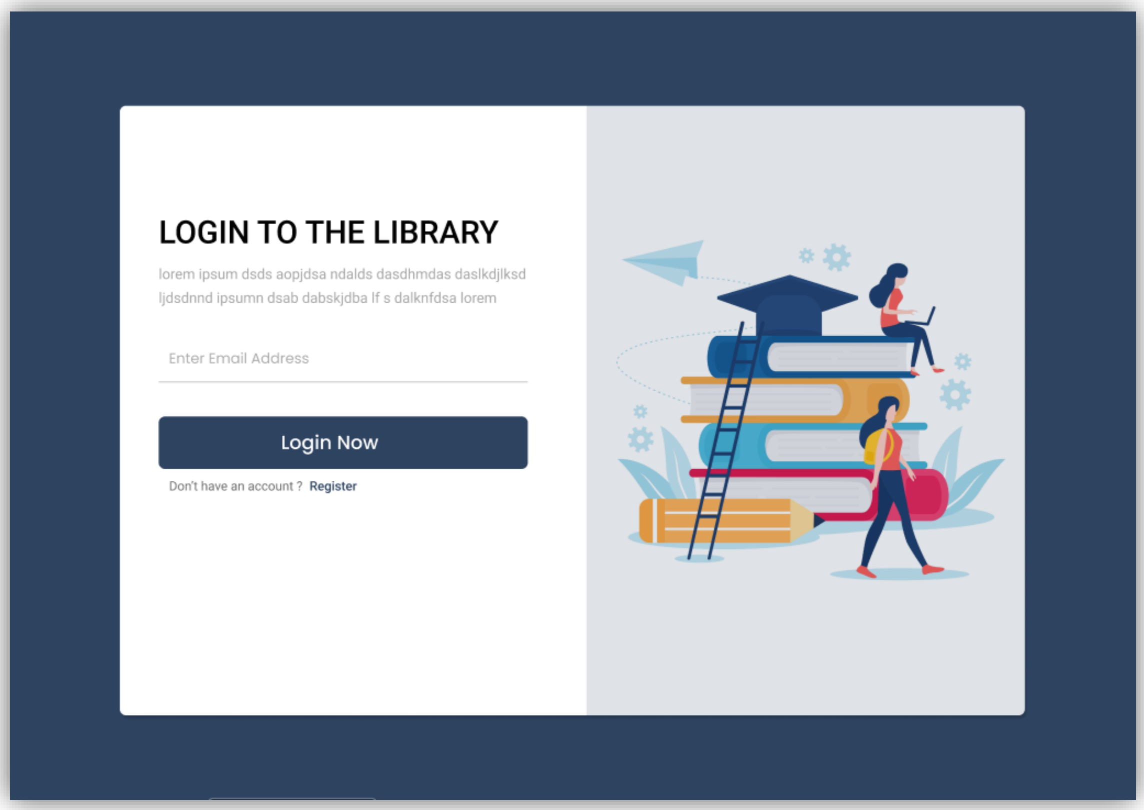
```go
// validate Admin
func AuthAdmin(c *gin.Context) {
    clientToken := c.Request.Header.Get("Authorization")
    if clientToken == "" {

        c.IndentedJSON(http.StatusInternalServerError, gin.H{"error": "no authorization token provided"})

        c.Abort()

        return

    }
    secret := []byte(os.Getenv("SECRET"))

    token, err := jwt.Parse(clientToken, func(token *jwt.Token) (interface{}, error) {
        if _, ok := token.Method.(*jwt.SigningMethodHMAC); !ok {
            return nil, fmt.Errorf("unexpected signing method: %v", token.Header["alg"])
        }

        return secret, nil
    })

    if err != nil {
        c.IndentedJSON(http.StatusInternalServerError, gin.H{"message": "error validating token", "error": err.Error()})
        c.Abort()
        return
    }

    if !token.Valid {
        c.IndentedJSON(http.StatusInternalServerError, gin.H{"message": "invalid token"})
        c.Abort()
        return
    }

    claims, _ := token.Claims.(jwt.MapClaims)
    fmt.Printf("claims %v", claims["role"])

```

Fig 1.3 - Validate Admin Function Snippet

2. LMS UI – Frontend:

The frontend of the Online Library Management System is developed using React.js along with custom CSS styles, ensuring a sleek and responsive user interface without relying on external CSS libraries. This approach offers flexibility, allowing for precise control over the design elements and layout of the application.

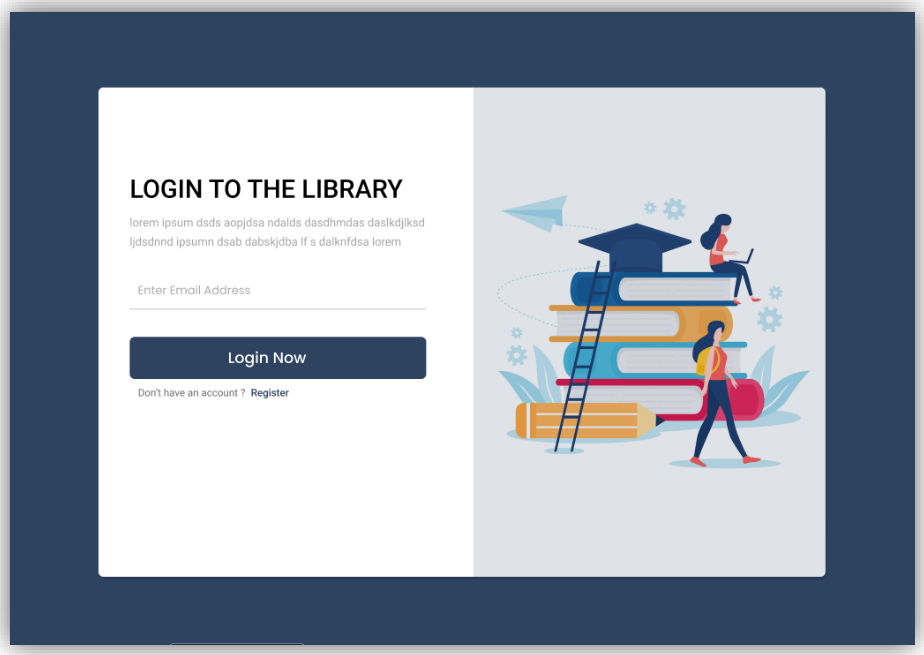
1. **React Js Components**: The frontend is structured using React.js components, facilitating modularity, reusability, and maintainability of code. Components such as Booklist, User Profile, and Signup Form are seamlessly integrated to create a cohesive user experience.
2. **State Management**: React.js state management mechanisms, including useState and useContext hooks, are utilized to manage application state efficiently. This enables dynamic rendering of content based on user interactions and data changes.
3. **Responsive Design**: Custom CSS styles are implemented to ensure a responsive design that adapts seamlessly to various screen sizes and devices. Media queries and flexbox/grid layouts are employed to optimize the user experience across desktops, tablets, and mobile devices..
4. **Custom Styling:** All visual elements, including typography, colors, and layout, are meticulously crafted using custom CSS styles. This approach provides full control over the design aesthetic, enabling the creation of a visually appealing and cohesive user interface.
5. **Performance Optimization**: Efforts are made to optimize the performance of the frontend, ensuring fast load times and smooth user interactions. Techniques such as code splitting, lazy loading, and memoization are employed to minimize unnecessary rendering and enhance overall responsiveness.
6. **Accessibility**: Accessibility considerations are integrated into the frontend design, ensuring that the application is usable by individuals with disabilities. Semantic HTML elements and ARIA attributes are leveraged to enhance screen reader compatibility and keyboard navigation.



LOGIN TO THE LIBRARY

lorem ipsum dsds aopjdsa ndalds dasdhmdas daslkdjlksd ljdsdnnd ipsumn dsab dabskjdba lf s dalknfdsa lorem

Fig 2.1 - Login Page



LOGIN TO THE LIBRARY

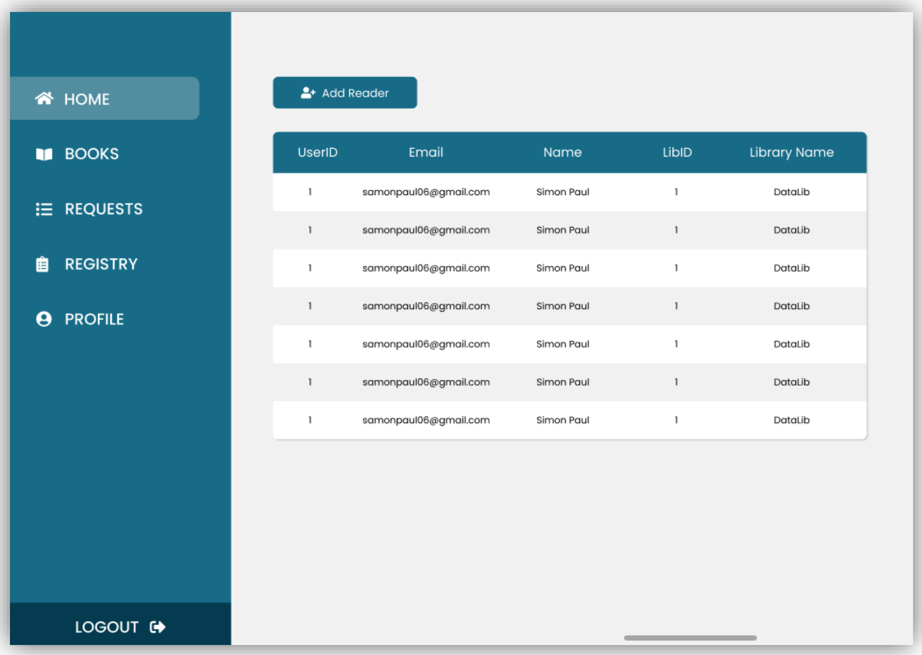
lorem ipsum dsds aopjdsa ndalds dasdhmdas daslkdjlksd ljdsdnnd ipsumn dsab dabskjdba lf s dalknfdsa lorem

Enter Email Address

Login Now

Don't have an account ? Register

Fig 2.2 - Register Page



UserID	Email	Name	LibID	Library Name
1	samonpaul06@gmail.com	Simon Paul	1	DataLib
1	samonpaul06@gmail.com	Simon Paul	1	DataLib
1	samonpaul06@gmail.com	Simon Paul	1	DataLib
1	samonpaul06@gmail.com	Simon Paul	1	DataLib
1	samonpaul06@gmail.com	Simon Paul	1	DataLib
1	samonpaul06@gmail.com	Simon Paul	1	DataLib
1	samonpaul06@gmail.com	Simon Paul	1	DataLib

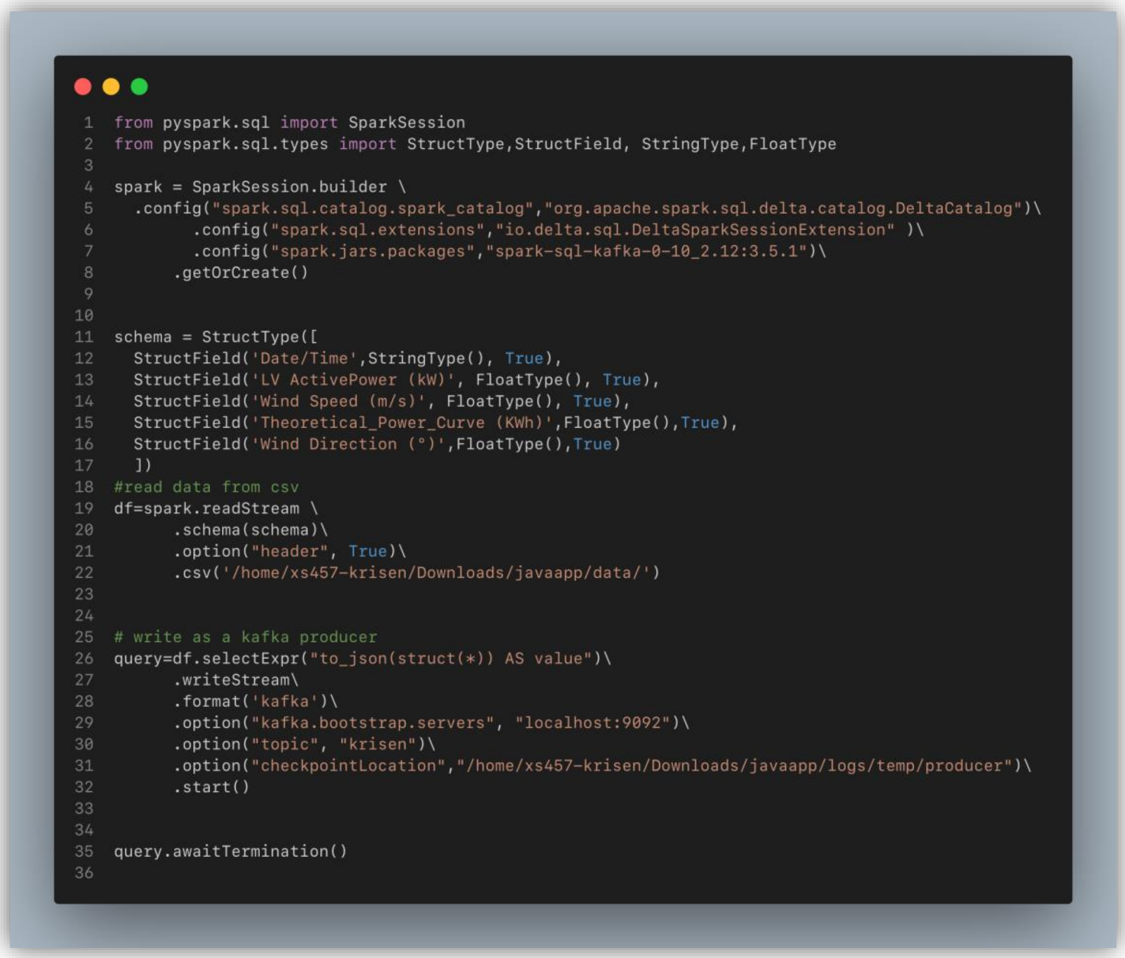
Fig 2.3 - Dashboard Page

3. Wind Power Generation Analysis using Kafka, Pyspark and Delta Table:

The project focuses on analyzing wind power generation data using advanced data engineering techniques. With the rapid rise of renewable energy sources, such as wind power, understanding the stochastic nature of their generation is crucial for effective integration into the power grid. This project aims to leverage Kafka, PySpark, and Delta Table to analyze and gain insights into the stochastic variation of wind power generation.

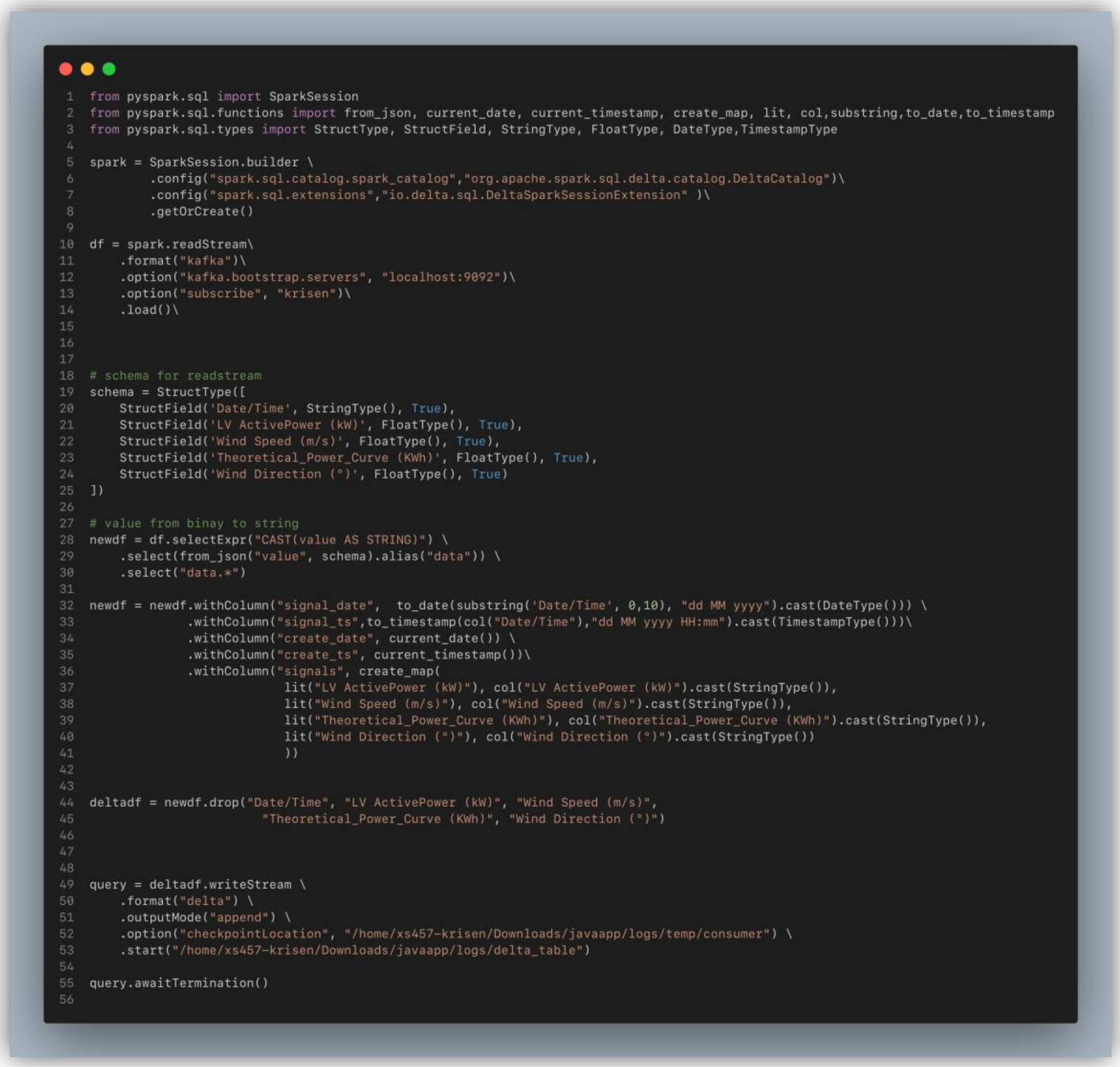
- **Project Description**:
  - **Renewable Energy and Wind Power**: Renewable energy sources, particularly wind energy, have emerged as viable alternatives to fossil fuels due to their sustainability and environmental benefits. Wind power generation capacity has witnessed significant growth in recent years and is considered a key contributor to the global transition towards cleaner energy solutions.
  - **Challenges In Wind Power Analysis**: Wind power generation exhibits stochastic and intermittent behavior, posing challenges for its integration into the power grid. Understanding the variations in wind power generation is essential for effectively managing energy resources and mitigating the impact of ramp events.
  - **Data Engineering Approach**: The project utilizes a data engineering approach to analyze wind power generation data. Kafka is employed for real-time data streaming, enabling the ingestion of high-volume data from wind turbines and other sources. PySpark, a powerful distributed computing framework, is used for processing and analyzing large-scale datasets efficiently. Delta Table, a reliable data lake storage layer, is leveraged for storing and managing the processed data, ensuring reliability, scalability, and ACID transactions.

- **Implementation Details**:
    - **Data Ingestion With Kafka**: Kafka is configured to ingest real-time data streams from wind turbines and other sources. The Kafka Connect framework is utilized to seamlessly integrate with various data sources and ensure continuous data ingestion.
    - **Data Processing With PySpark**: PySpark is employed to process and analyze the ingested wind power generation data. Distributed computing capabilities of PySpark enable efficient processing of large-scale datasets, including filtering, aggregation, and statistical analysis.
    - **Data storage with Delta Table**: Processed data is stored and managed using Delta Table, providing reliability, scalability, and transactional capabilities. Delta Table's versioning and transaction log features facilitate data lineage tracking and ensure data consistency and integrity.



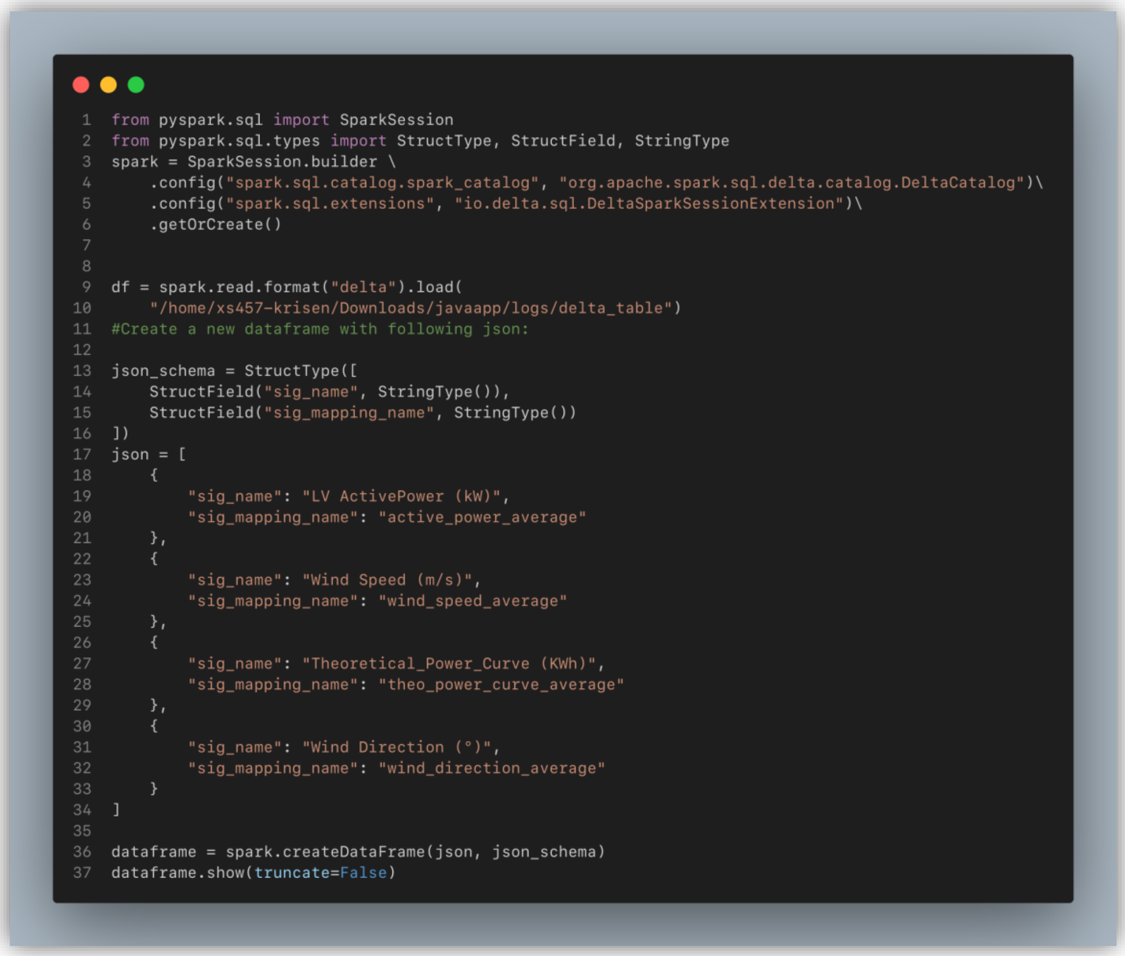
```python
from pyspark.sql import SparkSession
from pyspark.sql.types import StructType,StructField, StringType,FloatType

spark = SparkSession.builder \
  .config("spark.sql.catalog.spark_catalog","org.apache.spark.sql.delta.catalog.DeltaCatalog")\
        .config("spark.sql.extensions","io.delta.sql.DeltaSparkSessionExtension" )\
        .config("spark.jars.packages","spark-sql-kafka-0-10_2.12:3.5.1")\
      .getOrCreate()


schema = StructType([
  StructField('Date/Time',StringType(), True),
  StructField('LV ActivePower (kW)', FloatType(), True),
  StructField('Wind Speed (m/s)', FloatType(), True),
  StructField('Theoretical_Power_Curve (KWh)',FloatType(),True),
  StructField('Wind Direction (°)',FloatType(),True)
  ])
#read data from csv
df=spark.readStream \
      .schema(schema)\
      .option("header", True)\
      .csv('/home/xs457-krisen/Downloads/javaapp/data/')


# write as a kafka producer
query=df.selectExpr("to_json(struct(*)) AS value")\
      .writeStream\
      .format('kafka')\
      .option("kafka.bootstrap.servers", "localhost:9092")\
      .option("topic", "krisen")\
      .option("checkpointLocation","/home/xs457-krisen/Downloads/javaapp/logs/temp/producer")\
      .start()


query.awaitTermination()

```

Fig 3.1 - Writing Data to Kafka Topic using PySpark



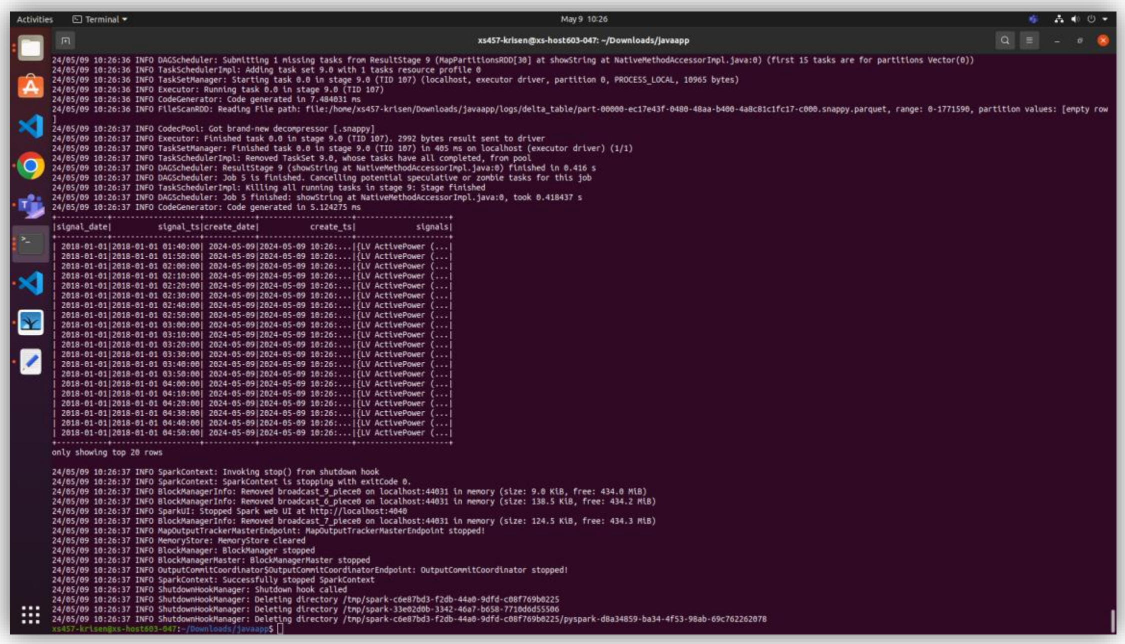
```python
from pyspark.sql import SparkSession
from pyspark.sql.functions import from_json, current_date, current_timestamp, create_map, lit, col,substring,to_date,to_timestamp
from pyspark.sql.types import StructType, StructField, StringType, FloatType, DateType,TimestampType

spark = SparkSession.builder \
        .config("spark.sql.catalog.spark_catalog","org.apache.spark.sql.delta.catalog.DeltaCatalog")\
        .config("spark.sql.extensions","io.delta.sql.DeltaSparkSessionExtension" )\
        .getOrCreate()

df = spark.readStream\
    .format("kafka")\
    .option("kafka.bootstrap.servers", "localhost:9092")\
    .option("subscribe", "krisen")\
    .load()\



# schema for readstream
schema = StructType([
    StructField('Date/Time', StringType(), True),
    StructField('LV ActivePower (kW)', FloatType(), True),
    StructField('Wind Speed (m/s)', FloatType(), True),
    StructField('Theoretical_Power_Curve (KWh)', FloatType(), True),
    StructField('Wind Direction (°)', FloatType(), True)
])

# value from binay to string
newdf = df.selectExpr("CAST(value AS STRING)") \
    .select(from_json("value", schema).alias("data")) \
    .select("data.*")

newdf = newdf.withColumn("signal_date",  to_date(substring('Date/Time', 0,10), "dd MM yyyy").cast(DateType())) \
             .withColumn("signal_ts",to_timestamp(col("Date/Time"),"dd MM yyyy HH:mm").cast(TimestampType()))\
             .withColumn("create_date", current_date()) \
             .withColumn("create_ts", current_timestamp())\
             .withColumn("signals", create_map(
                          lit("LV ActivePower (kW)"), col("LV ActivePower (kW)").cast(StringType()),
                          lit("Wind Speed (m/s)"), col("Wind Speed (m/s)").cast(StringType()),
                          lit("Theoretical_Power_Curve (KWh)"), col("Theoretical_Power_Curve (KWh)").cast(StringType()),
                          lit("Wind Direction (°)"), col("Wind Direction (°)").cast(StringType())
                          ))


deltadf = newdf.drop("Date/Time", "LV ActivePower (kW)", "Wind Speed (m/s)",
                       "Theoretical_Power_Curve (KWh)", "Wind Direction (°)")



query = deltadf.writeStream \
    .format("delta") \
    .outputMode("append") \
    .option("checkpointLocation", "/home/xs457-krisen/Downloads/javaapp/logs/temp/consumer") \
    .start("/home/xs457-krisen/Downloads/javaapp/logs/delta_table")

query.awaitTermination()

```

Fig 3.2 - Consuming Data from Kafka Topic



```python
from pyspark.sql import SparkSession
from pyspark.sql.types import StructType, StructField, StringType
spark = SparkSession.builder \
    .config("spark.sql.catalog.spark_catalog", "org.apache.spark.sql.delta.catalog.DeltaCatalog")\
    .config("spark.sql.extensions", "io.delta.sql.DeltaSparkSessionExtension")\
    .getOrCreate()


df = spark.read.format("delta").load(
    "/home/xs457-krisen/Downloads/javaapp/logs/delta_table")
#Create a new dataframe with following json:

json_schema = StructType([
    StructField("sig_name", StringType()),
    StructField("sig_mapping_name", StringType())
])
json = [
    {
        "sig_name": "LV ActivePower (kW)",
        "sig_mapping_name": "active_power_average"
    },
    {
        "sig_name": "Wind Speed (m/s)",
        "sig_mapping_name": "wind_speed_average"
    },
    {
        "sig_name": "Theoretical_Power_Curve (KWh)",
        "sig_mapping_name": "theo_power_curve_average"
    },
    {
        "sig_name": "Wind Direction (°)",
        "sig_mapping_name": "wind_direction_average"
    }
]

dataframe = spark.createDataFrame(json, json_schema)
dataframe.show(truncate=False)
```

Fig 3.3 - Creating DataFrame For Delta Table



```
24/05/09 10:26:36 INFO DAGScheduler: Submitting 1 missing tasks from ResultStage 9 (MapPartitionsRDD[30] at showString at NativeMethodAccessorImpl.java:0) (first 15 tasks are for partitions Vector(0))
24/05/09 10:26:36 INFO TaskSchedulerImpl: Adding task set 9.0 with 1 tasks resource profile 0
24/05/09 10:26:36 INFO TaskSetManager: Starting task 0.0 in stage 9.0 (TID 107) (localhost, executor driver, partition 0, PROCESS_LOCAL, 10965 bytes)
24/05/09 10:26:36 INFO Executor: Running task 0.0 in stage 9.0 (TID 107)
24/05/09 10:26:36 INFO CodeGenerator: Code generated in 7.484031 ms
24/05/09 10:26:36 INFO FileScanRDD: Reading File path: file:/home/xs457-krisen/Downloads/javaapp/logs/delta_table/part-00000-ec17e43f-0480-48aa-b400-4a8c81c1fc17-c000.snappy.parquet, range: 0-1771590, partition values: [empty row
]
24/05/09 10:26:37 INFO CodecPool: Got brand-new decompressor [.snappy]
24/05/09 10:26:37 INFO Executor: Finished task 0.0 in stage 9.0 (TID 107). 2992 bytes result sent to driver
24/05/09 10:26:37 INFO TaskSetManager: Finished task 0.0 in stage 9.0 (TID 107) in 405 ms on localhost (executor driver) (1/1)
24/05/09 10:26:37 INFO TaskSchedulerImpl: Removed TaskSet 9.0, whose tasks have all completed, from pool
24/05/09 10:26:37 INFO DAGScheduler: ResultStage 9 (showString at NativeMethodAccessorImpl.java:0) finished in 0.416 s
24/05/09 10:26:37 INFO DAGScheduler: Job 5 is finished. Cancelling potential speculative or zombie tasks for this job
24/05/09 10:26:37 INFO TaskSchedulerImpl: Killing all running tasks in stage 9: Stage finished
24/05/09 10:26:37 INFO DAGScheduler: Job 5 finished: showString at NativeMethodAccessorImpl.java:0, took 0.418437 s
24/05/09 10:26:37 INFO CodeGenerator: Code generated in 5.124275 ms
+----------+-------------------+-----------+--------------------+--------------------+
|signal_date|          signal_ts|create_date|           create_ts|             signals|
+----------+-------------------+-----------+--------------------+--------------------+
| 2018-01-01|2018-01-01 01:40:00| 2024-05-09|2024-05-09 10:26:...|{LV ActivePower (...|
| 2018-01-01|2018-01-01 01:50:00| 2024-05-09|2024-05-09 10:26:...|{LV ActivePower (...|
| 2018-01-01|2018-01-01 02:00:00| 2024-05-09|2024-05-09 10:26:...|{LV ActivePower (...|
| 2018-01-01|2018-01-01 02:10:00| 2024-05-09|2024-05-09 10:26:...|{LV ActivePower (...|
| 2018-01-01|2018-01-01 02:20:00| 2024-05-09|2024-05-09 10:26:...|{LV ActivePower (...|
| 2018-01-01|2018-01-01 02:30:00| 2024-05-09|2024-05-09 10:26:...|{LV ActivePower (...|
| 2018-01-01|2018-01-01 02:40:00| 2024-05-09|2024-05-09 10:26:...|{LV ActivePower (...|
| 2018-01-01|2018-01-01 02:50:00| 2024-05-09|2024-05-09 10:26:...|{LV ActivePower (...|
| 2018-01-01|2018-01-01 03:00:00| 2024-05-09|2024-05-09 10:26:...|{LV ActivePower (...|
| 2018-01-01|2018-01-01 03:10:00| 2024-05-09|2024-05-09 10:26:...|{LV ActivePower (...|
| 2018-01-01|2018-01-01 03:20:00| 2024-05-09|2024-05-09 10:26:...|{LV ActivePower (...|
| 2018-01-01|2018-01-01 03:30:00| 2024-05-09|2024-05-09 10:26:...|{LV ActivePower (...|
| 2018-01-01|2018-01-01 03:40:00| 2024-05-09|2024-05-09 10:26:...|{LV ActivePower (...|
| 2018-01-01|2018-01-01 03:50:00| 2024-05-09|2024-05-09 10:26:...|{LV ActivePower (...|
| 2018-01-01|2018-01-01 04:00:00| 2024-05-09|2024-05-09 10:26:...|{LV ActivePower (...|
| 2018-01-01|2018-01-01 04:10:00| 2024-05-09|2024-05-09 10:26:...|{LV ActivePower (...|
| 2018-01-01|2018-01-01 04:20:00| 2024-05-09|2024-05-09 10:26:...|{LV ActivePower (...|
| 2018-01-01|2018-01-01 04:30:00| 2024-05-09|2024-05-09 10:26:...|{LV ActivePower (...|
| 2018-01-01|2018-01-01 04:40:00| 2024-05-09|2024-05-09 10:26:...|{LV ActivePower (...|
| 2018-01-01|2018-01-01 04:50:00| 2024-05-09|2024-05-09 10:26:...|{LV ActivePower (...|
+----------+-------------------+-----------+--------------------+--------------------+
only showing top 20 rows

24/05/09 10:26:37 INFO SparkContext: Invoking stop() from shutdown hook
24/05/09 10:26:37 INFO SparkContext: SparkContext is stopping with exitCode 0.
24/05/09 10:26:37 INFO BlockManagerInfo: Removed broadcast_9_piece0 on localhost:44031 in memory (size: 9.0 KiB, free: 434.0 MiB)
24/05/09 10:26:37 INFO BlockManagerInfo: Removed broadcast_6_piece0 on localhost:44031 in memory (size: 138.5 KiB, free: 434.2 MiB)
24/05/09 10:26:37 INFO SparkUI: Stopped Spark web UI at http://localhost:4040
24/05/09 10:26:37 INFO BlockManagerInfo: Removed broadcast_7_piece0 on localhost:44031 in memory (size: 124.5 KiB, free: 434.3 MiB)
24/05/09 10:26:37 INFO MapOutputTrackerMasterEndpoint: MapOutputTrackerMasterEndpoint stopped!
24/05/09 10:26:37 INFO MemoryStore: MemoryStore cleared
24/05/09 10:26:37 INFO BlockManager: BlockManager stopped
24/05/09 10:26:37 INFO BlockManagerMaster: BlockManagerMaster stopped
24/05/09 10:26:37 INFO OutputCommitCoordinator$OutputCommitCoordinatorEndpoint: OutputCommitCoordinator stopped!
24/05/09 10:26:37 INFO SparkContext: Successfully stopped SparkContext
24/05/09 10:26:37 INFO ShutdownHookManager: Shutdown hook called
24/05/09 10:26:37 INFO ShutdownHookManager: Deleting directory /tmp/spark-c6e87bd3-f2db-44a0-9dfd-c08f769b0225
24/05/09 10:26:37 INFO ShutdownHookManager: Deleting directory /tmp/spark-33e02d0b-3342-46a7-b658-7710d6d55506
24/05/09 10:26:37 INFO ShutdownHookManager: Deleting directory /tmp/spark-c6e87bd3-f2db-44a0-9dfd-c08f769b0225/pyspark-d8a34859-ba34-4f53-98ab-69c762262078
xs457-krisen@xs-host603-047:~/Downloads/javaapp$ 
```

Fig 3.4 - Kafka Topic Data Output

4. Web Testing Automation Using Selenium

The project focuses on automating web testing processes using Selenium, a powerful tool for browser automation. With the increasing complexity of web applications and the need for efficient testing methodologies, Selenium provides a robust framework for automating repetitive testing tasks and ensuring the quality and reliability of web applications.

Project Description:

- Significance of Web Testing:

    - Web testing plays a crucial role in ensuring the functionality, performance, and usability of web applications across different browsers and platforms.
    - Manual testing processes can be time-consuming, error-prone, and resource-intensive, especially for large-scale web applications.
- Automation with selenium:
    - Selenium is a widely-used open-source tool for automating web browsers, enabling testers to simulate user interactions and verify application behavior programmatically.
    - The project utilizes Selenium's capabilities to automate various testing scenarios, including functional testing, regression testing, and cross-browser testing.
- Testing Scenarios:
    - The project encompasses a range of testing scenarios tailored to the specific requirements of the web application under test.
    - Common testing scenarios include user authentication, form submission, navigation, data validation, and error handling.

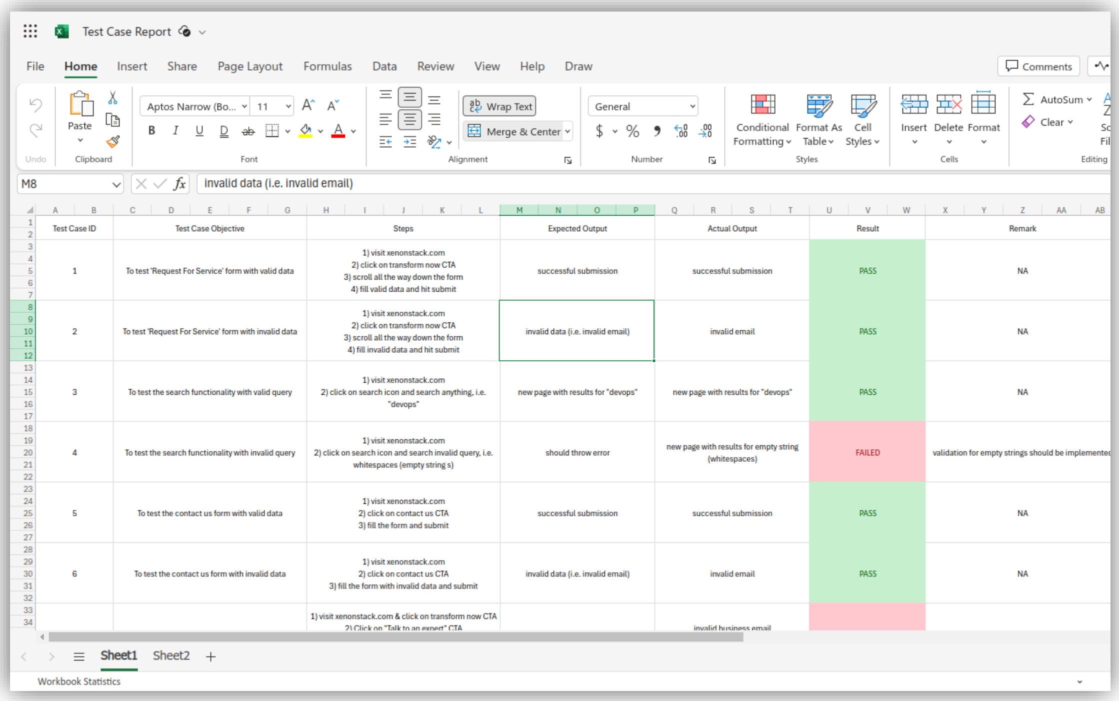
Implementation Details:

- Selenium Webdriver:
    - Selenium WebDriver, the core component of Selenium, is used to interact with web browsers programmatically.
    - WebDriver provides a rich set of APIs for navigating web pages, interacting with web elements, and executing JavaScript code.
- The Automation Framework:
    - A test automation framework is developed using Selenium WebDriver to organize and manage test scripts efficiently.

    - The framework includes reusable components, such as page objects, test utilities, and configuration files, to streamline test development and maintenance.
- Test Script Development:
    - Test scripts are developed using programming languages such as Java, Python, or C#, depending on the project requirements.
    - Test scripts leverage Selenium WebDriver APIs to simulate user interactions, verify expected outcomes, and report test results.

Benefits of Selenium Testing:

- Efficiency and Productivity:
    - Selenium automation reduces the time and effort required for manual testing, enabling testers to focus on more critical aspects of quality assurance.
    - Automated tests can be executed repeatedly and consistently across different environments, browsers, and configurations.
- Accuracy and Reliability:
    - Selenium tests execute predefined test scenarios with precision, minimizing the risk of human errors and ensuring reliable test results.
    - Automated tests provide immediate feedback on application behavior, allowing for timely identification and resolution of issues.


Test Case Report

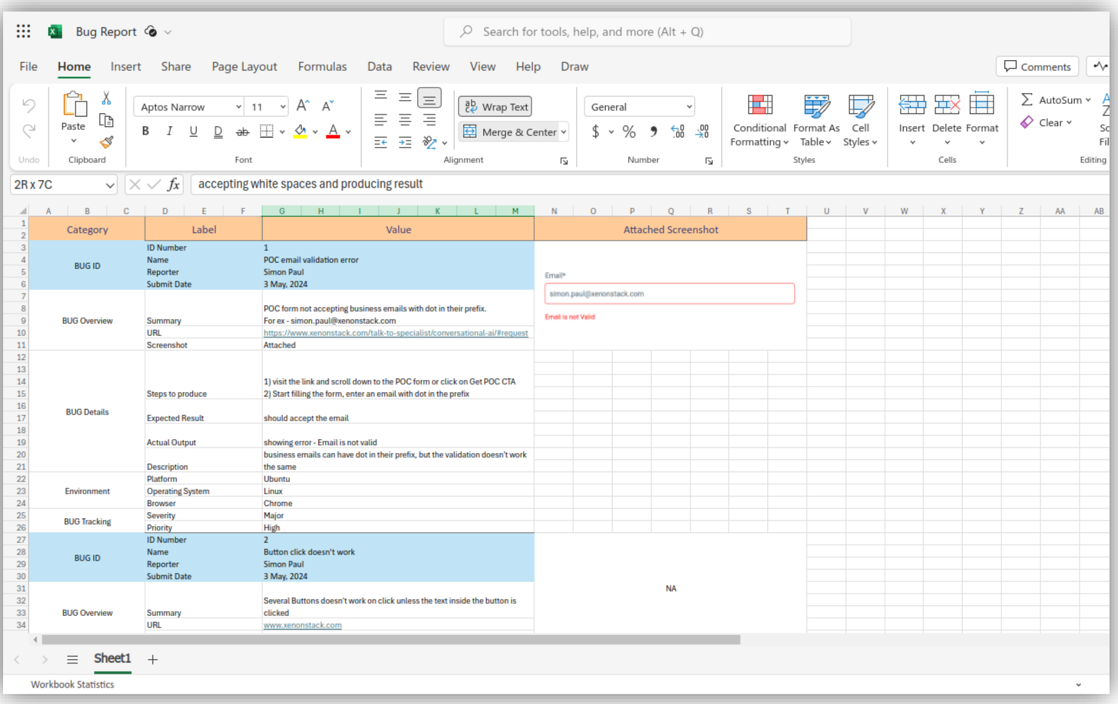
M8 invalid data (i.e. invalid email)

Test Case ID	Test Case Objective	Steps	Expected Output	Actual Output	Result	Remark
1	To test 'Request For Service' form with valid data	1) visit xenonstack.com 2) click on transform now CTA 3) scroll all the way down the form 4) fill valid data and hit submit	successful submission	successful submission	PASS	NA
2	To test 'Request For Service' form with invalid data	1) visit xenonstack.com 2) click on transform now CTA 3) scroll all the way down the form 4) fill invalid data and hit submit	invalid data (i.e. invalid email)	invalid email	PASS	NA
3	To test the search functionality with valid query	1) visit xenonstack.com 2) click on search icon and search anything, i.e. "devops"	new page with results for "devops"	new page with results for "devops"	PASS	NA
4	To test the search functionality with invalid query	1) visit xenonstack.com 2) click on search icon and search invalid query, i.e. whitespaces (empty string s)	should throw error	new page with results for empty string (whitespaces)	FAILED	validation for empty strings should be implemented
5	To test the contact us form with valid data	1) visit xenonstack.com 2) click on contact us CTA 3) fill the form and submit	successful submission	successful submission	PASS	NA
6	To test the contact us form with invalid data	1) visit xenonstack.com 2) click on contact us CTA 3) fill the form with invalid data and submit	invalid data (i.e. invalid email)	invalid email	PASS	NA
		1) visit xenonstack.com & click on transform now CTA 2) Click on "Talk to an expert" CTA		invalid business email		

Sheet1 Sheet2

Workbook Statistics

Fig 4.1 - Test Case Report



2R x 7C | accepting white spaces and producing result

Category	Label	Value	Attached Screenshot
BUG ID	ID Number	1	Email* simon.paul@xenonstack.com Email is not Valid
	Name	POC email validation error	
	Reporter	Simon Paul	
	Submit Date	3 May, 2024	
BUG Overview	Summary	POC form not accepting business emails with dot in their prefix. For ex - simon.paul@xenonstack.com	
	URL	https://www.xenonstack.com/talk-to-specialist/conversational-ai/#request	
	Screenshot	Attached	
BUG Details	Steps to produce	1) visit the link and scroll down to the POC form or click on Get POC CTA 2) Start filling the form, enter an email with dot in the prefix	
	Expected Result	should accept the email	
	Actual Output	showing error - Email is not valid	
	Description	business emails can have dot in their prefix, but the validation doesn't work the same	
Environment	Platform	Ubuntu	
	Operating System	Linux	
	Browser	Chrome	
BUG Tracking	Severity	Major	
	Priority	High	
BUG ID	ID Number	2	NA
	Name	Button click doesn't work	
	Reporter	Simon Paul	
	Submit Date	3 May, 2024	
BUG Overview	Summary	Several Buttons doesn't work on click unless the text inside the button is clicked	
	URL	www.xenonstack.com	

Fig 4.2 - Bug Report File

5. Containerization With Docker & Kubernetes

The project focuses on implementing containerization solutions using Docker and Kubernetes to streamline application deployment, management, and scalability. By leveraging containerization technologies, the project aims to enhance the efficiency, reliability, and scalability of software deployment processes in modern cloud-native environments.

- **Project Description:**
  - Containerization and its significance:
    1. Containerization technologies such as Docker provide a lightweight and portable solution for packaging applications and their dependencies into isolated containers.

    2. Containerization offers several advantages, including improved resource utilization, faster deployment times, and increased consistency across different environments.

  - Role of Docker & Kubernetes:
    1. Docker simplifies the process of creating, distributing, and running containerized applications, providing a consistent runtime environment across development, testing, and production environments.
    2. Kubernetes, an open-source container orchestration platform, enables automated deployment, scaling, and management of containerized applications, facilitating efficient resource utilization and ensuring high availability and reliability.

  - Project Objectives:
    1. The project aims to containerize a sample application using Docker containers and deploy it to a Kubernetes cluster.
    2. Key objectives include achieving seamless application deployment, managing containerized services efficiently, and implementing scalability and fault tolerance mechanisms.

- **Implementation Details:**
  - Containerization with Docker:
    1. The sample application is containerized using Docker containers, which encapsulate the application code, dependencies, and runtime environment.
    2. Dockerfiles are used to define the configuration and dependencies of the containerized application, ensuring consistency and reproducibility across different environments.

  - Orchestration With Kubernetes:

1. Kubernetes is utilized for orchestrating the deployment, scaling, and management of containerized applications within a Kubernetes cluster.
2. Kubernetes resources such as Pods, Deployments, Services, and Ingress are used to define and manage the desired state of the application, ensuring optimal performance and reliability.

- Scalability and High Availability:
    1. Kubernetes enables automatic scaling of application instances based on resource demand, ensuring efficient resource utilization and maintaining application performance under varying workloads.
    2. High availability and fault tolerance are achieved through Kubernetes features such as Pod replication, health checks, and automated failover mechanisms.

- **Benefits Of Containerization:**
    - Portability and Flexibility:
        1. Containerization enables the creation of portable and self-contained application units that can be deployed across different infrastructure environments, including on-premises data centers and cloud platforms.
        2. Docker containers provide a consistent runtime environment, ensuring that applications behave predictably regardless of the underlying infrastructure.
    - Efficiency And Resource Utilization:
        1. Containerized applications consume fewer resources compared to traditional virtual machines, leading to improved resource utilization and cost efficiency.

2. Kubernetes enables efficient scheduling and distribution of application workloads across cluster nodes, optimizing resource allocation and maximizing utilization.

Fig 5.1 - Docker Image

Fig 5.2 Kubernetes Logo

My internship at Xenonstack Pvt. Ltd. Mohali commenced on **January 25th, 2024,** and is scheduled to conclude on **July 25th, 2024**.

Fig – 3.1 Xenonstack Operation and tools

The **Initial two months** were dedicated to comprehensive training, providing me with a solid foundation in database, api, cloud, linux and the tools I utilized.

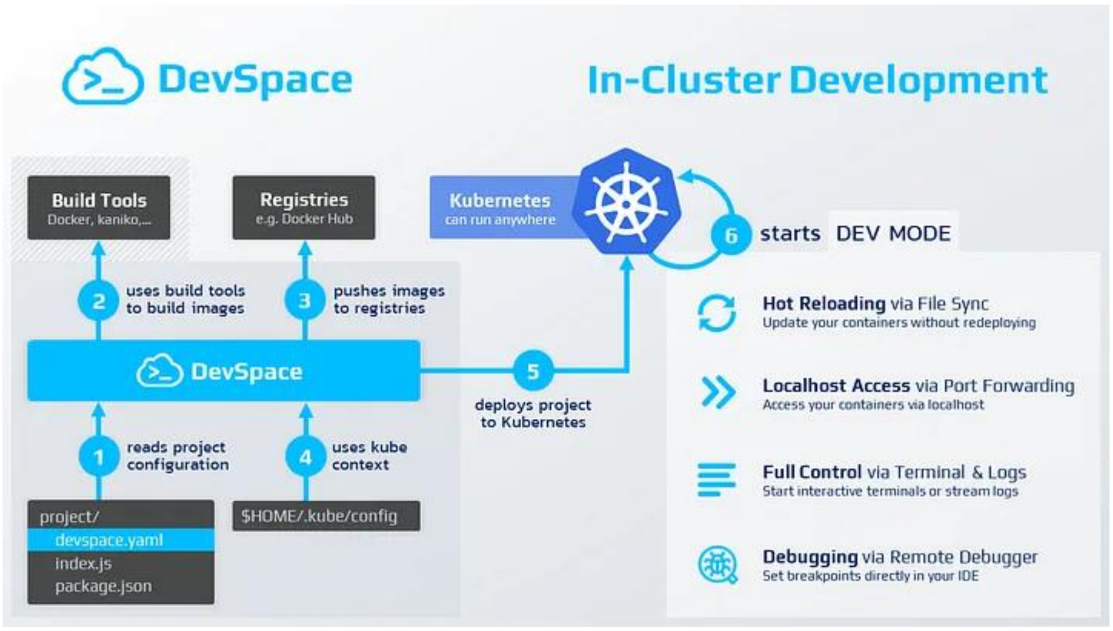

DevSpace
In-Cluster Development
Build Tools
Docker, kaniko,...
Registries
e.g. Docker Hub
Kubernetes
can run anywhere
2 uses build tools to build images
3 pushes images to registries
6 starts DEV MODE
DevSpace
5 deploys project to Kubernetes
1 reads project configuration
4 uses kube context
project/
devspace.yaml
index.js
package.json
$HOME/.kube/config
Hot Reloading via File Sync
Update your containers without redeploying
Localhost Access via Port Forwarding
Access your containers via localhost
Full Control via Terminal & Logs
Start interactive terminals or stream logs
Debugging via Remote Debugger
Set breakpoints directly in your IDE

Fig 3.2 – Technology overview given at Xenonstack.

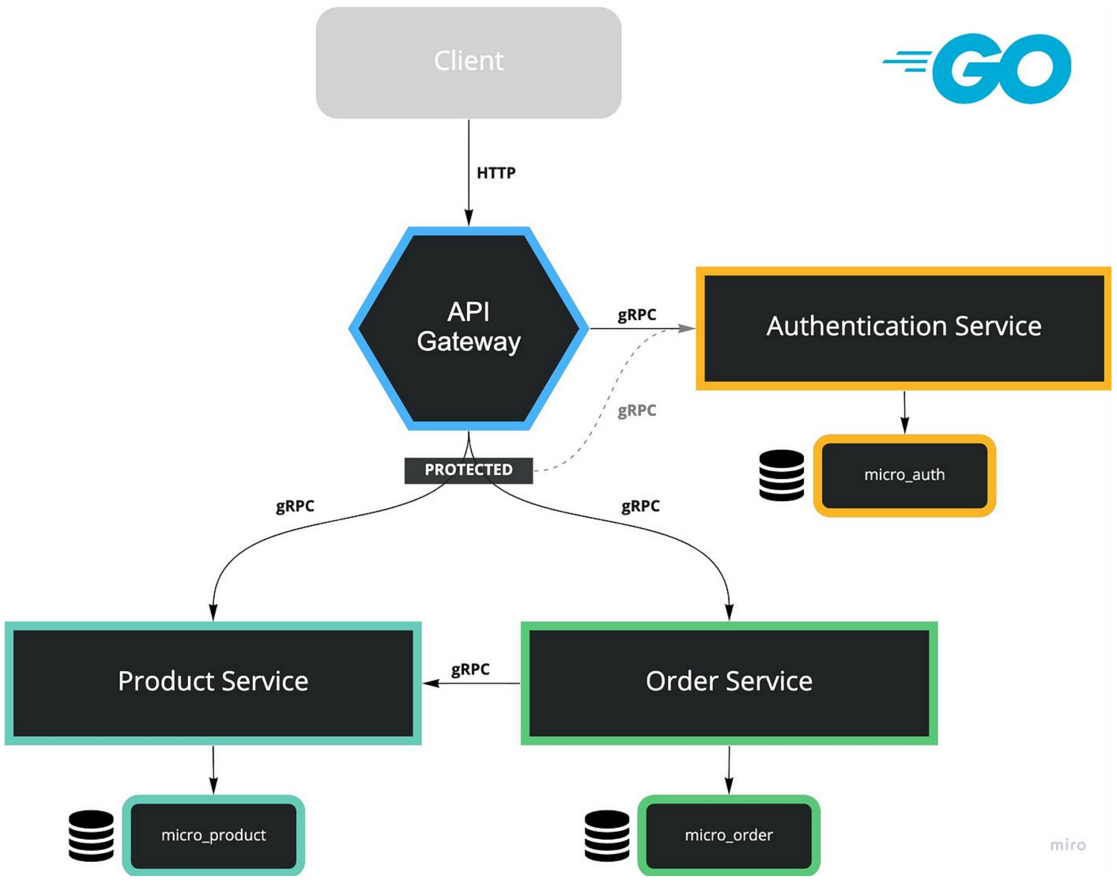
1. **GoLang**

1. GoLang, also known as Go, is an open-source programming language. It is a statically typed language created at Google in 2009. It is widely used by developers to simplify workflows, create defect-free code, and minimize risks associated with software development. It was designed to be simple, efficient

and easy to learn, making it a popular choice for building scalable networks, web applications and command-line tools. Here are some key points about Go:

2. **Purpose**: Go is designed to make development simple and produce efficient code. It is a feature rich programming language that is known for its concurrency and inbuilt garbage collection, making it a popular choice among developers.
3. **Features**:
    1. **Simplicity**: Go is known for its simplistic approach to programming yet produces efficient results.
    2. **Concurrency Support**: Concurrency is the ability to deal with multiple things at once, but no two things are happening at the same instance. Go provides a rich support to concurrency.
    3. **Powerful Standard Library**: Go has a very powerful standard library, which in turn contains a rich set of packages that developers use.

In summary, Golang acts as the backend technology which held the Api in real time world problems and very useful if used correctly.



2. React js

React makes it painless to create interactive UIs. Design simple views for each state in your application, and React will efficiently update and render just the right components when your data changes.

Declarative views make your code more predictable and easier to debug.

Build encapsulated components that manage their own state, then compose them to make complex UIs.

Since component logic is written in JavaScript instead of templates, you can easily pass rich data through your app and keep state out of the DOM.

We don’t make assumptions about the rest of your technology stack, so you can develop new features in React without rewriting existing code.

React can also render on the server using Node and power mobile apps using React Native.

1. **Purpose and Features**:

- Currently, ReactJS is gaining quick popularity as the best JavaScript framework among web developers. It is playing an essential role in the front-end ecosystem. The important features of ReactJS are as follows.
- JSX
- Components
- One-way Data Binding
- Virtual DOM
- Simplicity
- Performance

2. **Advantages**:

- It is composable.
- It is declarative.
- Write once and learn anywhere.
- It is simple.
- SEO friendly.
- Fast, efficient, and easy to learn.
- It guarantees stable code.
- It is backed by a strong community.

3. **Structure of reactjs**:

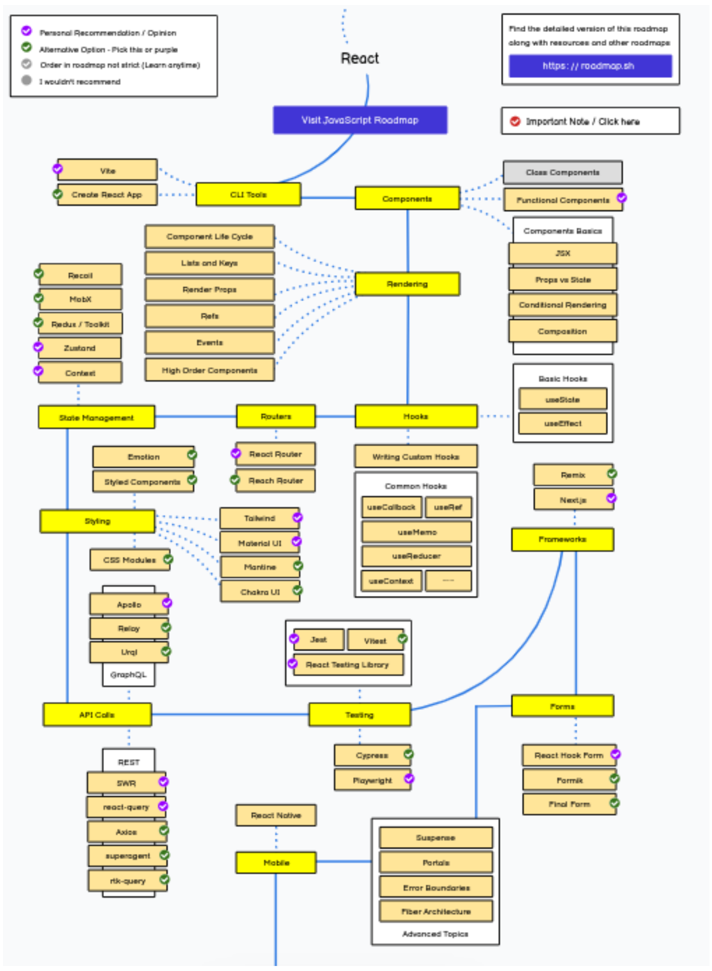


Fig 3.0 – Technology overview given at Xenonstack.

3. Data Engineering:

Data engineering is the practice of designing and building systems for collecting, storing, and analyzing data at scale. It is a broad field with applications in just about every industry. Organizations can collect massive amounts of data, and they need the right people and technology to ensure it is in a highly usable state by the time it reaches data scientists and analysts.

In addition to making the lives of data scientists easier, working as a data engineer can allow you to make a tangible difference in a world where we'll be producing 463 exabytes per day by 2025. That's one and 18 zeros of bytes worth of data. Fields like machine learning and deep learning can’t succeed without data engineers to process and channel that data.

1. **Definition**:
    - Data engineering is the practice of designing and building systems for collecting, storing, and analyzing data at scale. It is a broad field with applications in just about every industry.
    - It encompasses various layers, including network security, data encryption, access controls, and monitoring
2. **Challenges**:
    - **Shared Responsibility Model**:
        - Data security is a joint effort between the provider and the user.
        - Users must configure security settings correctly.
    - **Data Breaches and Misconfigurations**:
        - Misconfigured permissions can lead to data exposure.
        - Regular security assessments are crucial.
    - **Emerging Threats**:
        - As kafka adoption grows, new threats emerge (e.g., serverless vulnerabilities, supply chain attacks).
3. **Best Practices**:
    - **Least Privilege**: Grant minimal permissions to users and services.
    - **Regular Backups**: Ensure data resilience.
    - **Patch Management**: Keep software up to date.
    - **Security Training**: Educate users about security practices.

In summary, cloud security involves a holistic approach, combining technical controls, policies, and user awareness.

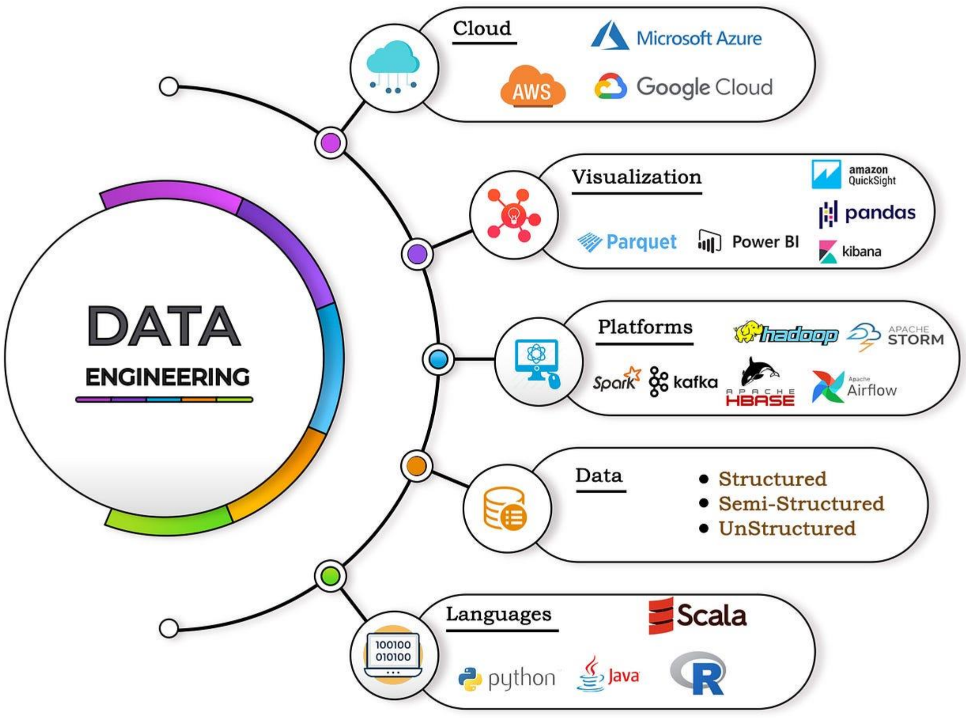


Fig 3.1 – Technology overview given at Xenonstack.

4. Cloud:

While Docker is a container runtime, Kubernetes is a platform for running and managing containers from many container runtimes. Kubernetes supports numerous container runtimes including Docker, container, CRI-O, and any implementation of the Kubernetes CRI (Container Runtime Interface). A good metaphor is Kubernetes as an “operating system” and Docker containers are “apps” that you install on the “operating system”.

On its own, Docker is highly beneficial to modern application development. It solves the classic problem of “works on my machine” but then nowhere else. The container orchestration tool Docker Swarm can handle a production container workload deployment of a few containers. When a system grows and needs to add many containers networked to each other, standalone Docker can face some growing pains that Kubernetes helps address.

When comparing the two, a better comparison is of Kubernetes with Docker Swarm. Docker Swarm, or Docker swarm mode, is a container orchestration tool like Kubernetes, meaning it allows the management of multiple containers deployed across multiple hosts running the Docker server. Swarm mode is disabled by default and is something that needs to be setup and configured by a DevOps team.

Kubernetes orchestrates clusters of machines to work together and schedules containers to run on those machines based on their available resources. Containers are grouped together, through declarative definition, into pods, which is the basic unit of Kubernetes. Kubernetes automatically manages things like service discovery, load balancing, resource allocation, isolation, and scaling your pods vertically or horizontally. It has been embraced by the open-source community and is now part of the Cloud Native Computing Foundation. Amazon, Microsoft, and Google all offer managed Kubernetes services on their cloud computing platforms, which significantly reduces the operational burden of running and maintaining Kubernetes clusters and their containerized workloads.

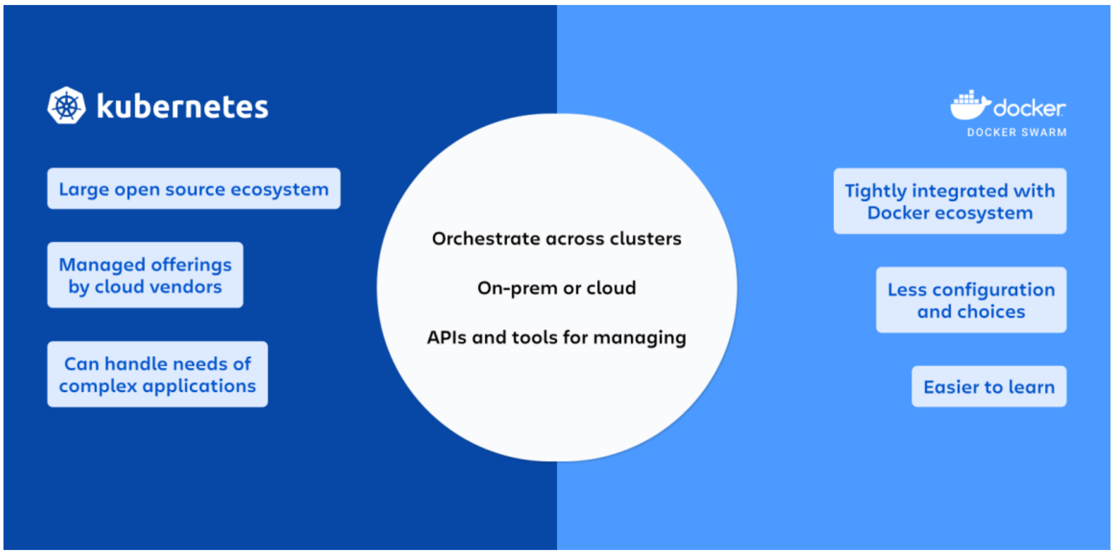

kubernetes
Large open source ecosystem
Managed offerings by cloud vendors
Can handle needs of complex applications
Orchestrate across clusters
On-prem or cloud
APIs and tools for managing
docker
DOCKER SWARM
Tightly integrated with Docker ecosystem
Less configuration and choices
Easier to learn

Fig 3.2 – Technology overview given at Xenonstack.

5. Software Testing:

Software testing is a process of identifying the correctness of software by considering

its all attributes (Reliability, Scalability, Portability, Re-usability, Usability) and evaluating the execution of software components to find the software bugs or errors or defects.
Software testing provides an independent view and objective of the software and gives surety of fitness of the software. It involves testing of all components under the required services to confirm whether it is satisfying the specified requirements or not. The process also provides the client with information about the quality of the software.

Testing is mandatory because it will be a dangerous situation if the software fails at any time due to lack of testing. So, without testing software cannot be deployed to the end user.

Testing is a group of techniques to determine the correctness of the application under the predefined script but, testing cannot find all the defects of the application. The main intent of testing is to detect failures of the application so that failures can be discovered and corrected. It does not demonstrate that a product functions properly under all conditions but only that it is not working in some specific conditions.

Testing furnishes comparison that compares the behavior and state of software against mechanisms because the problem can be recognized by the mechanism. The mechanism may include past versions of the same specified product, comparable products, and interfaces of expected purpose, relevant standards, or other criteria but not limited up to these.

Testing includes an examination of code and also the execution of code in various environments and conditions as well as all the examining aspects of the code. In the current scenario of software development, a testing team may be separate from the development team so that Information derived from testing can be used to correct the process of software development.

The success of software depends upon acceptance of its targeted audience, easy graphical user interface, strong functionality load test, etc. For example, the audience of banking is totally different from the audience of a video game. Therefore, when an organization develops a software product, it can assess whether the software product will be beneficial to its purchasers and other audiences.

Chapter 4: Project methodology

A **project management methodology** is a structured approach that guides how projects are planned, executed, monitored, and controlled. Different methodologies offer various principles, processes, and tools to help ensure successful project delivery. An Agile methodology will be followed for this project, emphasizing iterative development, continuous improvement, and flexibility. The project will be divided into sprints with clear deliverables and milestones.

1. **Agile**:
    - **Description**: Agile is a flexible and collaborative approach to project management. It emphasizes iterative development, adaptability, and customer feedback.
    - **Key Characteristics**:
        - Collaboration: Cross-functional teams work together closely.
        - Iterative Process: Projects are broken down into smaller iterations (sprints) with frequent deliverables.
        - Adaptability: Changes are welcomed throughout the project.
        - Customer Involvement: Regular feedback from stakeholders.
    - **Common Methodologies Used with Agile**:
        - Scrum: Time-boxed sprints with specific roles (Scrum Master, Product Owner, Development Team).
        - Kanban: Visualize work on a board with columns (e.g., “To Do,” “In Progress,” “Done”).
        - Extreme Programming (XP): Focuses on software development practices (pair programming, continuous integration).
        - Crystal: Tailored to the team’s size and project complexity.
        - Scrumban: A mix of Scrum and Kanban.

    - Suitable for: Software development, product development, and cross-functional teams.
2. **Waterfall**:
    - Description: Waterfall follows a linear sequence of phases, where each phase must be completed before moving to the next.
    - **Key Characteristics**:
        - Sequential Process: Requirements, design, implementation, testing, deployment.
        - Predictable Timeline: Detailed planning upfront.
        - Clear Dependencies: Tasks are interlinked.
    - Suitable for: Large projects with well-defined requirements and stable environments.
3. **Scrum**:
    - Description: Scrum is an Agile framework with short sprints (usually 2-4 weeks).
    - **Key Characteristics**:
        - Sprints: Time-bound iterations with specific goals.
        - Daily Stand-ups: Brief daily meetings.
        - Product Backlog: Prioritized list of features.
        - Sprint Backlog: Tasks for the current sprint.
    - Suitable for: Software development, where flexibility and adaptability are crucial.

4 Project Flow

The project will follow a phased approach:

Data Analysis Process and its essential steps. Whether you’re working on a project or making data-driven decisions for your organization, understanding this process is crucial. Here are the key steps:

1. **Define the Problem or Research Question**:

Begin by clearly defining the problem or question you want to address through data analysis. What insights are you seeking? What business challenge are you trying to solve?

- **Problem Statement**:
  - The problem statement succinctly describes the issue or challenge that needs to be addressed. It should be clear, concise, and specific.
  - Example: "Our company's website experiences slow loading times during peak hours."
- **Research Question**:
  - A research question guides your investigation and helps you focus on relevant aspects. It should be answerable through research and analysis.
  - Example: "What factors contribute to the slow loading times on our website, and how can we optimize performance?"
- **Key Considerations**:
  - **Scope**: Define the boundaries of your problem. Is it related to a specific area, process, or system?
  - **Context**: Understand the context in which the problem exists. Consider organizational goals, user needs, and constraints.
  - **Measurability**: Ensure that the problem or research question can be measured or evaluated objectively.
  - **Feasibility**: Assess whether it's feasible to address the problem within available resources (time, budget, expertise).
- **Example (Website Performance)**:
  - **Problem Statement**: "Our e-commerce website experiences slow loading times, leading to high bounce rates and reduced sales."
  - **Research Question**: "What specific factors (such as server response time, image optimization, or third-party scripts) contribute to the slow loading times, and how can we improve website performance?"
  -

2. **Collect Data**:

Gather relevant data from various sources. This could include databases, spreadsheets, APIs, or other data repositories. Ensure data quality and completeness.

- **Define Data Requirements**:
    - Clearly outline what data you need. Consider:
        - **Types of Data**: Is it numerical, categorical, or textual?
        - **Sources**: Where can you find the data? (e.g., databases, surveys, APIs)
        - **Volume**: How much data is required?
- **Data Collection Methods**:
    - **Surveys and Questionnaires**: Gather information directly from participants.
    - **Observations**: Observe and record data (e.g., behavior, events).
    - **Existing Databases**: Utilize publicly available datasets or proprietary databases.
    - **Web Scraping**: Extract data from websites.
    - **Sensor Data**: Collect data from sensors (e.g., temperature, humidity).
    - **Interviews**: Conduct one-on-one or group interviews.
- **Data Quality Assurance**:
    - Ensure data accuracy, completeness, and consistency.
    - Handle missing values and outliers appropriately.
    - Validate data against predefined criteria.
- **Ethical Considerations**:
    - Respect privacy and confidentiality.
    - Obtain informed consent when collecting personal data.
    - Anonymize sensitive information.
- **Data Storage and Organization**:
    - Choose a suitable format (e.g., CSV, JSON, database).
    - Organize data into tables or structures.
    - Backup data to prevent loss.
- **Data Validation and Cleaning**:
    - Validate data entries against predefined rules.

    - Clean data by removing duplicates, correcting errors, and standardizing formats.
- **Document the Data Collection Process**:
    - Create a data collection plan.
    - Record details such as sampling methods, data sources, and any modifications made during collection.

3. **Data Cleaning**:

Cleanse and preprocess the data. Handle missing values, outliers, and inconsistencies. Transform data into a usable format.

- **Handling Missing Values**:
    - **Identify Missing Data**:
        - Check for null values (NaN or blanks) in your dataset.
        - Understand why data is missing (e.g., random, systematic, or non-response).
    - **Strategies for Handling Missing Data**:
        - **Imputation**: Fill missing values with estimates (mean, median, mode, or regression-based imputation).
        - **Deletion**: Remove rows or columns with missing data (use cautiously).
        - **Advanced Techniques**: Consider using machine learning models to predict missing values.
- **Removing Duplicates**:
    - Identify and remove duplicate records from your dataset.
    - Be cautious not to accidentally remove valid data.
- **Standardizing Formats**:
    - Ensure consistent formats for categorical data (e.g., capitalization, abbreviations).
    - Normalize numerical data (e.g., scaling features to a common range).
- **Handling Outliers**:
    - Detect outliers (extreme values) that may skew your analysis.

    - Decide whether to remove, transform, or keep outliers based on domain knowledge.
- **Correcting Inconsistencies**:
    - Check for inconsistent data (e.g., misspellings, variations in encoding).
    - Use tools like regular expressions or string matching to standardize text data.
- **Dealing with Noisy Data**:
    - Noise refers to irrelevant or erroneous data.
    - Review data for anomalies or unexpected patterns.
- **Feature Engineering**:
    - Create new features from existing ones (e.g., extracting date components, combining variables).
    - Feature engineering can enhance model performance.
- **Domain-Specific Cleaning**:
    - Understand the context of your data (e.g., industry-specific rules, business logic).
    - Clean data based on domain expertise.

4. **Analyzing the Data**:

Apply statistical and quantitative methods to explore patterns, relationships, and trends within the data.

Common techniques include:

- **Descriptive Statistics**: Summarize and describe data.
- **Inferential Statistics**: Make predictions or draw conclusions based on a sample of data.
- **Hypothesis Testing**: Assess whether observed differences are statistically significant.
- **Regression Analysis**: Understand relationships between variables.
- **Cluster Analysis**: Group similar data points.
- **Factor Analysis**: Identify underlying factors.

- **Time Series Analysis**: Analyze data over time.
- **Text Analysis**: Extract insights from text data.
- **Neural Networks** and **Data Mining**: Advanced techniques for complex problems.
- **Defining Objectives**: Before diving into data analysis, it's crucial to understand what you want to achieve. Clearly define your goals and the questions you want to answer. For example, are you trying to understand customer behavior, optimize marketing strategies, or predict disease outbreaks? These objectives guide the entire analysis process.
- **Collecting Data**: Next, gather relevant data from various sources. This data can be structured (like databases or spreadsheets) or unstructured (like text or images). The quality and quantity of data significantly impact the analysis results.
- **Data Cleaning**: Raw data often contains errors, missing values, or inconsistencies. Data cleaning involves removing duplicates, filling in missing values, and ensuring data consistency. Clean data is essential for accurate analysis.
- **Data Transformation**: Transform the data into a suitable format for analysis. This may involve aggregating data, creating new variables, or normalizing values. Common techniques include scaling, encoding categorical variables, and feature engineering.
- **Exploratory Data Analysis (EDA)**: Explore the data visually and statistically. EDA helps you understand patterns, relationships, and distributions. Techniques include histograms, scatter plots, and summary statistics.
- **Modeling and Analysis Techniques**: Apply statistical or machine learning models to the data. Techniques vary based on the problem and data type. Examples include linear regression, decision trees, clustering, and neural networks.
- **Interpreting Results**: Analyze model outputs and interpret the findings. What do the coefficients mean? How well does the model perform? Are there significant patterns or trends?

- **Data Storytelling**: Communicate your findings effectively. Create visualizations, reports, or presentations to convey insights to stakeholders. Storytelling bridges the gap between data analysis and decision-making.

5. **Data Visualization**:

Create visual representations of your findings. Charts, graphs, and plots help communicate insights effectively.

Examples include bar charts, scatter plots, line graphs, heatmaps, and more.

1. **Types of Data Visualizations**:
   - **Bar Charts**: These display categorical data using rectangular bars. They're great for comparing values across different categories.
   - **Line Charts**: Ideal for showing trends over time. They connect data points with lines.
   - **Pie Charts**: Represent parts of a whole. Each slice corresponds to a category or percentage.
   - **Scatter Plots**: Show the relationship between two variables. Each point represents an observation.
   - **Heatmaps**: Visualize data using color intensity. Useful for showing patterns in matrices.
   - **Histograms**: Display the distribution of continuous data.
   - **Box Plots**: Illustrate the spread and central tendency of data.
   - **Network Graphs**: Depict relationships between entities (nodes) and connections (edges).
2. **Tools for Data Visualization**:
   - **Matplotlib**: A popular Python library for creating static, interactive plots.
   - **Seaborn**: Built on Matplotlib, Seaborn provides high-level functions for attractive visualizations.

   - **D3.js**: A powerful JavaScript library for creating interactive web-based visualizations.
   - **Tableau**: A user-friendly tool for creating professional visualizations without coding.
   - **Power BI**: Microsoft’s business intelligence tool for data visualization.

Chapter 5: Project Challenges

The technologies practiced during the internship present a unique set of challenges compared to traditional on-premises technologies. Here are some of the key hurdle’s organizations face:

- **Shared Responsibility Model:** Cloud providers are responsible for the security of the underlying infrastructure, but the security of the data and applications themselves remains the customer's responsibility. This shared model can lead to confusion and require clear delineation of who is accountable for what.
- **Lack of Visibility and Control:** Organizations often lack full visibility into their cloud environment, especially in multi-cloud deployments. This can make it difficult to monitor for suspicious activity, enforce security policies consistently, and identify misconfigurations that could create vulnerabilities.
- **Data Security and Privacy:** Data breaches and unauthorized access to sensitive information are major concerns in the cloud. Organizations need to ensure data is encrypted at rest and in transit, and that access controls are properly configured. Additionally, compliance with data privacy regulations like GDPR and CCPA adds another layer of complexity.

- **Complexity of Cloud Environments:** Multi-cloud and hybrid cloud deployments can significantly increase the complexity of managing security. Organizations need to ensure consistent security policies and configurations across all their cloud environments.
- **API Security:** APIs (Application Programming Interfaces) are essential for connecting applications and services in the cloud. However, APIs can also be a target for attackers. Organizations need to secure their APIs with strong authentication and authorization mechanisms.

By understanding these challenges, organizations can develop a comprehensive cloud security strategy that mitigates risks and protects their valuable data and applications in the cloud.

Chapter 6: Conclusion and Future Scope

My internship journey at Xenonstack, particularly within the various technology, has been a fulfilling and enlightening experience. In this concluding chapter, I will delve into how the internship objectives were achieved, the array of skills acquired, the notable results and observations gleaned from my time at the company, and the challenges encountered along the way.

6.1 Meeting Project Objectives

This internship's main objective was to state the main objective of your internship related to various technology. This objective was achieved through the following accomplishments:

- Learning cloud, docker, Kubernetes.
- Bash, PowerShell scripting.
- Golang
- React js
- Data engineering

- Software testing

By successfully completing these tasks, I directly contributed to gaining practical insights into these technologies and apply theoretical knowledge acquired during academic studies. This objective was successfully achieved through hands-on experiences, collaboration with industry professionals, and participation in various projects aimed at enhancing cloud security posture. By actively engaging in tasks related to configuring security tools like docker hub, golang, kafka, pyspark, I gained a deeper understanding of clouds and their real-world applications. Additionally, involvement in planning, execution, and monitoring of tasks provided valuable insights into the complexities of securing cloud environments.

6.2 Skill Development

Throughout the internship, I had the opportunity to develop and refine both scientific and professional skills, which are invaluable for my future career endeavors.

6.3 Results, Observations, and Work Experiences

i. During my tenure at Xenonstack, I had the privilege to witness firsthand how cloud security operations are conducted within a leading organization. Some notable results, observations, and work experiences include:
ii. Observing the day-to-day operations of the various tasks, including backend, ui, devops and the testing of the official site of Xenonstack.
iii. Collaborating with different IT teams, which gain insights into various aspects of cloud infrastructure, software engineering, UI, Data Engineering and Software Testing.
iv. Participating in Saturday workshop meetings and discussions, where I gained insights into emerging threats, industry best practices, and the company's overall strategy.
v. These experiences provided me with a well-rounded understanding of various technologies within a real-world organizational setting.

6.4 Challenges Faced

i. No learning journey is devoid of challenges, and my internship at Emerson was no exception. Some of the challenges I encountered during the internship include:
ii. Grappling with the complexity of some of the technologies, which necessitated continuous learning and adaptation to stay abreast of evolving threats and technologies.
iii. Balancing the need for learning new concepts with the demands of completing assigned tasks within the stipulated timelines.
iv. Adapting to a new professional environment and acclimatizing to the company culture, which required proactive communication and a willingness to learn from colleagues.
v. Despite these challenges, each hurdle presented a valuable learning opportunity, fostering resilience, resourcefulness, and adaptability, which are essential qualities for success in the dynamic field of these technologies.

References

https://go.dev/tour/list

https://gorm.io/

https://github.com/gin-gonic/gin

https://blogs.halodoc.io/golang-unit-testing/

https://roadmap.sh/react

https://react.dev/learn

https://www.w3schools.com/react/react_getstarted.asp

https://www.w3schools.com/nodejs/nodejs_npm.asp

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Regular_expressions

https://kafka.apache.org/quickstart

https://sparkbyexamples.com/pyspark-tutorial/

https://subhamkharwal.medium.com/pyspark-structured-streaming-read-from-kafka-64c40767155f https://docs.databricks.com/en/delta/index.html

https://medium.com/@venkatkarthick15/how-to-create-delta-lake-tables-9490bbca7449

https://delta.io/blog/write-kafka-stream-to-delta-lake/

https://selenium-python.readthedocs.io/getting-started.html

https://www.browserstack.com/guide/python-selenium-to-run-web-automation-test

https://docs.docker.com/get-started/overview/

https://kubernetes.io/docs/concepts/