Project Design Phase-II
Technology Stack (Architecture & Stack)

Date	12 october 2023
Team ID	NM2023TMID06078
Project Name	Quantitative Analysis of Candidate in 2019 LokSabha Elections
Maximum Marks	4 Marks

Technical Architecture:

The Deliverable shall include the architectural diagram as below and the information as per the table1 & table 2

Example: Order processing during pandemics for offline mode

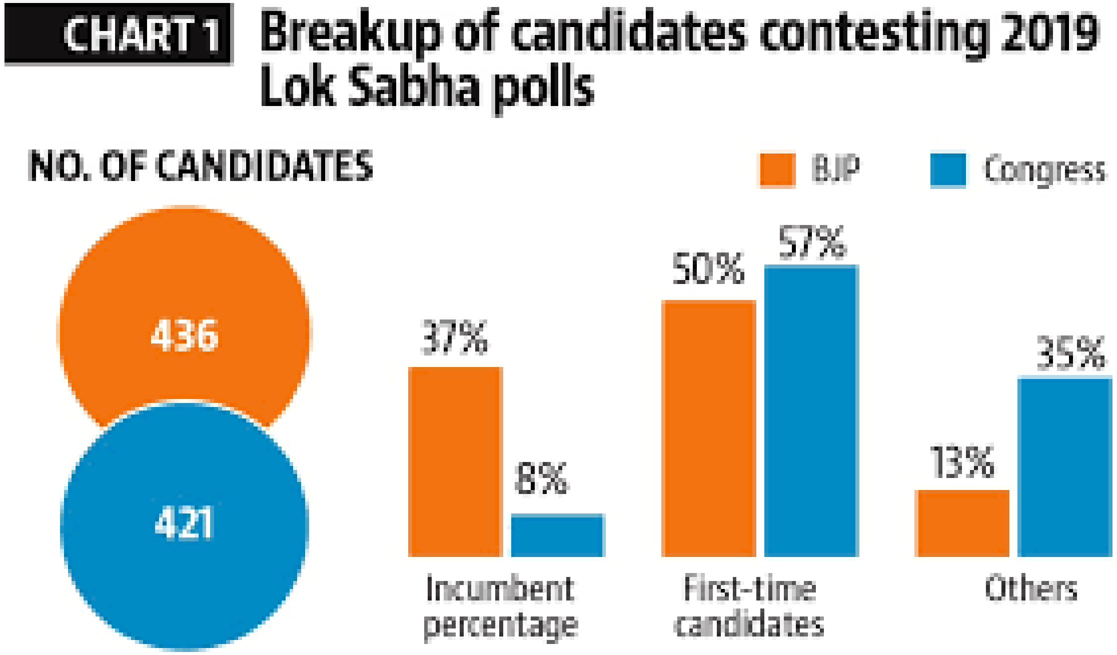

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	The user interface for data entry, visualization, and reporting related to the candidate's campaign and election data.	HTML, CSS, JavaScript, React.js, D3.js
2.	Application Logic-1	Core application logic responsible for data processing, analytics, and generating key quantitative insights.	Python (with libraries like Pandas, NumPy), Node.js
3.	Application Logic-2	Additional application logic for specific data collection, data cleaning, and preparation tasks	Java, Ruby, Scala
4.	Application Logic-3	More application logic for advanced statistical analysis, machine learning model integration, and predictive modeling.	R, Python (with machine learning libraries), Golang
5.	Database	A relational database for storing structured data related to candidate profiles, election results, and voter demographics.	PostgreSQL, MySQL, SQL Server
6.	Cloud Database	Scalable and cloud-hosted database for real-time data storage, analytics, and reporting.	Amazon RDS, Google Cloud SQL, Azure SQL Database
7.	File Storage	Repository for unstructured data such as images, documents, and external datasets used for analysis.	Amazon S3, Google Cloud Storage, Azure Blob Storage
8.	External API-1	Integration with external data sources like election commission data, polling data, and campaign finance reports.	REST API, GraphQL, Web scraping
9.	External API-2	Additional third-party service integration for data enrichment, sentiment analysis, and social media engagement metrics.	Twitter API, Facebook Graph API, Google Maps API
10.	Machine Learning Model	Utilization of machine learning algorithms for sentiment analysis, predictive modeling, and voter behavior prediction.	Tenso rFlow, scikit-learn, PyTorch
11.	Infrastructure (Server / Cloud)	Server/Cloud hosting environment for deploying the application, handling data pipelines, and scaling as needed.	Amazon Web Services, Google Cloud Platform, Azure

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Utilized open-source libraries and frameworks to build the application, enhancing development efficiency and reducing costs.	Django, Flask, Ruby on Rails, Angular, React
2.	Security Implementations	Implementation of security measures to protect sensitive election data and ensure data privacy and integrity.	OAuth, JWT, SSL/TLS, Role-Based Access Control.
3.	Scalable Architecture	Designing an architecture that allows the application to scale horizontally or vertically to accommodate increased data and user loads.	Microservices, Docker, Kubernetes, Load Balancers
4.	Availability	Ensuring high availability and uptime of the application to allow continuous data analysis and access for users.	Redundancy, Failover, Cloud Load Balancing
5.	Performance	Optimizing the system's performance to handle large datasets, complex calculations, and concurrent user interactions efficiently.	Caching (Redis, Memcached), CDN, Database Indexing, Load Testing