Summary of Jobseeker portal

A report submitted in partial fulfillment of the requirements for

**Summer Internship**

by

**Siddhartha Singhal**

Under the guidance of

**Shri Raj Kumar Gupta**

(Technical Director, IT),

NIC, Lucknow



**National Informatics Centre**

Sarojini Naidu Marg, Yojna Bhawan, Lucknow, Uttar Pradesh 226001, India

August, 2025

# **Undertaking**

I declare that the work presented in this report titled “Summary of Jobseeker portal”, submitted to the National Informatics Centre, Lucknow, for the certificate of **Summer Internship, 2025**, is my original work. I have not plagiarized or submitted the same work of any other internship program.

August, 2025

Lucknow (Siddhartha Singhal)

# **Content**

**Acknowledgement**

**Abstract**

1. **Introduction**  
   1.1 About Jobseeker Portal  
   1.2 Importance of Jobseeker Portals  
   1.3 Objective of the Project  
   1.4 Scope of the Analysis
2. **Tools and Technologies Used**  
   2.1 Python  
   2.2 SQL  
   2.3 Power BI  
   2.4 Supporting Libraries (Pandas, Matplotlib, Seaborn)
3. **Data Collection and Understanding**  
   3.1 Data Source  
   3.2 Description of Columns  
   3.3 Sample Records
4. **Data Preprocessing**  
   4.1 Handling Null Values  
   4.2 Data Type Conversion  
   4.3 Feature Engineering  
   4.4 Outlier Detection and Handling
5. **SQL-Based Data Analysis**  
   5.1 Daily Registration Averages  
   5.2 Registration Completion Timelines  
   5.3 Duplicate Email Binning  
   5.4 Verified vs. Unverified Users
6. **Visualization Using Power BI**  
   6.1 Dashboard Design  
   6.2 Filters and Slicers  
   6.3 Key Charts and Metrics  
   6.4 User Interaction and Storytelling
7. **Key Insights and Observations**  
   7.1 User Engagement

7.2 Profile Completion & Verification  
7.3 Technical & Browser Usage  
7.4 Growth Trends  
7.5 Data Gaps

1. **Conclusion**

# **Acknowledgement**

I would like to express my deepest gratitude Mr. Raj Kumar Gupta sir for providing me the opportunity to work on such a large project. His trust in my abilities has not only expanded my technical knowledge but also challenged me to think critically and push my boundaries.

And I would like to extend my gratitude to Mr. Pradeep sir for their constant guidance and support throughout the course of this project. Their expertise in the field, encouraging words and insightful feedback played a vital role in the successful completion of this work. Their mentorship helped me in enhancing my technical skills and also inspired me to pursue innovative solutions and think critically.

I would also like to extend my sincere thanks to NIC Lucknow, for providing the resources and facilities necessary for conducting this research. The support and facilities offered by the institute were crucial part in the successful realization of this project. I would also like to acknowledge my family and friends for their encouragement and motivation at different times.

Thank you all for your support and engagement.

**-Siddhartha Singhal**

# **Abstract**

This project presents a comprehensive analytical study conducted as part of a data analytics training program. The objective was to gain hands-on experience with real-world business scenarios using realistic, large-scale test data representing the behavior of jobseekers. The focus was to understand user behavior, engagement patterns, and registration trends using data science techniques and tools such as Python, SQL, and Power BI.

The dataset, designed to mimic real-world platform data, included information such as registration timestamps, login activity, verification status, browser usage, and user engagement metrics. Extensive data cleaning and preprocessing were performed to ensure data quality and prepare it for analysis. Key features were engineered, including account age, login patterns, and verification indicators.

Exploratory Data Analysis (EDA) uncovered key trends: a considerable proportion of users did not complete the verification process, over 40% never logged in post-registration, and there were multiple instances of shared email usage. Additionally, delays in final profile submissions indicated potential usability or engagement challenges.

Interactive dashboards built in Power BI visualized these findings effectively. Key visuals included regional comparisons, time-series trends in registration, and onboarding funnel drop-offs. These helped in identifying areas of friction and improvement opportunities.

Based on these insights, the study proposed actionable recommendations such as simplifying the onboarding workflow to promote same-day completion, sending timely verification reminders, and implementing safeguards like CAPTCHA and two-factor authentication to prevent misuse and increase platform trust.

In summary, this project demonstrates how data science can be applied to evaluate digital service platforms. The insights gained through data exploration and visualization can inform better decision-making and enhance user experience — a critical skill set for any aspiring data analyst.

# **Chapter 1: Introduction**

## 1.1 About Jobseeker Portal

This project is based on a simulated scenario designed to reflect real-world challenges faced by government-led job matching portals. The hypothetical platform acts as a bridge between job seekers and job providers, offering features such as user registration, job listings, login-based activity, and profile verification.

The simulated portal mimics the structure of many digital employment systems in India, especially those aiming to improve employment access for youth in rural and semi-urban areas. Employers (public and private) can post jobs, and job seekers can browse opportunities and participate in recruitment events. This use case was selected to reflect how digital platforms can be leveraged for employment facilitation and how data analytics can uncover gaps in their usability and engagement.

## 1.2 Importance of Jobseeker Portals

In today’s digital economy, such portals serve a crucial role in connecting job seekers with suitable opportunities by eliminating geographical, procedural, and informational barriers. They simplify recruitment processes, provide timely access to job listings, and enhance overall transparency.

For policymakers and administrators, these platforms also offer a scalable method to collect and analyze user behavior data. Insights derived from such data can shape evidence-based policy interventions and platform enhancements. Advanced portals further use AI-powered job recommendations and analytics to personalize user experiences. Given the size and diversity of India’s workforce, well-designed job portals are essential tools for bridging labor market gaps and empowering job seekers.

## 1.3 Objective of the Project

The main objective of this training project was to simulate a real-world data analytics scenario by working with a realistic dummy dataset modeled on jobseeker’s portal activity. Using this dataset, the project aimed to apply the core tools and techniques of data science and business intelligence. The key objectives were:

* To analyze registration, login, and verification behavior.
* To identify trends in user engagement and drop-offs.
* To detect anomalies such as shared emails or delayed profile completion.
* To visualize trends using Power BI dashboards for better interpretation.
* To provide data-driven recommendations for improving platform usability.

This hands-on project helped develop the practical skills required to evaluate public service platforms and propose improvements based on empirical evidence.

## 1.4 Scope of the Analysis

It focuses on a test dataset created to represent user behavior over a defined period. The key areas covered in this project include:

* Data cleaning and feature engineering using Python and Pandas.
* SQL-based querying for behavioral insights and anomaly detection.
* Power BI visualizations to present findings to non-technical stakeholders.

This analysis does not cover the employer-side journey, external job market factors, or predictive analytics such as churn modeling. However, these areas are highlighted as potential extensions for future projects. The insights produced are meant to improve onboarding, verification, and engagement within a government-style digital employment platform.

# **Chapter 2: Tools and Technologies Used**

Understanding and leveraging the right tools is essential for conducting efficient, scalable, and insightful data analysis. This project utilized a set of powerful tools and technologies designed for data preprocessing, transformation, visualization, and interpretation. The main technologies used include **Python**, **SQL**, **Power BI**, and supporting Python libraries such as **Pandas**, **Matplotlib**, and **Seaborn**. Each tool served a distinct purpose in the end-to-end workflow, from data ingestion to generating business intelligence insights.

## 2.1 Python

**Python** is a versatile, high-level programming language that played a central role in this project. It was used for:

* **Data Cleaning**: Handling missing values, type conversion, outlier detection, and data formatting.
* **Data Transformation**: Creating derived columns, normalizing datasets, parsing dates, and feature engineering.
* **Automation**: Automating repetitive tasks such as filtering and reporting.

Python’s readability and large ecosystem make it especially suitable for data analysis and machine learning tasks. In this project, it acted as the backbone for data preprocessing, allowing robust and efficient handling of structured data.

Furthermore, its integration with powerful libraries like Pandas, NumPy, Matplotlib, and Seaborn made it an ideal environment to perform initial exploratory data analysis (EDA) before transferring data to visualization platforms.

## 2.2 SQL

**SQL (Structured Query Language)** was employed for interacting with structured databases. It was instrumental in:

* **Data Extraction**: Pulling relevant subsets of data from relational databases.
* **Data Filtering**: Applying conditions, joins, aggregations, and groupings to extract meaningful patterns.
* **Preprocessing at Source**: Performing transformations (e.g., CASE statements, type casting) at the database level to reduce data handling overhead in Python.

SQL enabled efficient manipulation of large datasets stored in relational databases like Microsoft SQL Server (SSMS). Queries were used not only to filter and join data but also to create temporary views and perform conditional aggregations critical for KPI computation and trend detection.

## 2.3 Power BI

**Power BI** is Microsoft’s interactive data visualization and business intelligence tool. It was used for the final step in the pipeline:

* **Dashboard Creation**: Interactive dashboards were created to display insights in a user-friendly and dynamic manner.
* **Visual Storytelling**: Graphs, slicers, KPIs, and maps were used to represent trends, patterns, and anomalies.
* **Real-time Analysis**: With the capability to connect directly to databases or refreshed datasets, Power BI provided a near real-time view of key metrics.

Power BI helped in bridging the gap between raw data and decision-making. Its drag-and-drop interface combined with DAX (Data Analysis Expressions) made it easy to create calculated measures and hierarchies. It empowered both technical and non-technical stakeholders to interact with the data through filters, drill-downs, and interactive visuals.

## 2.4 Supporting Libraries (Pandas, Matplotlib, Seaborn)

Python’s data ecosystem was enriched through the use of **supporting libraries**, each serving a unique function:

* **Pandas**: Crucial for data manipulation and cleaning. DataFrames allowed for indexing, filtering, merging, and transforming tabular data with ease. It was used extensively for EDA and feature engineering.
* **Matplotlib**: A foundational library used for plotting static graphs like line plots, bar charts, and histograms. It provided control over visual elements like labels, grids, and plot customization.
* **Seaborn**: Built on top of Matplotlib, Seaborn enabled the creation of more visually appealing and complex statistical plots. It was ideal for creating correlation heatmaps, box plots, distribution plots, and multi-variable visualizations.

Together, these libraries formed the core of Python’s visualization and data handling capabilities, enabling seamless exploration, validation, and storytelling through visual data representation before moving to BI tools.

# **Chapter 3: Data Collection and Understanding**

The foundation of any successful data analysis project lies in a thorough understanding of the dataset being used. This section outlines the source from which the data was obtained, a detailed description of the dataset’s structure, and a few representative sample records. This step was essential in ensuring that further analysis, modeling, or visualization was grounded in a clear comprehension of the dataset’s content, quality, and relevance.

## 3.1 Data Source

The dataset used in this project was a simulated test dataset designed to mirror real-world user behavior on a government employment portal. It was created as part of a data analytics training exercise to provide hands-on experience with data cleaning, exploration, and visualization workflows.

The original data was structured in a Microsoft Excel workbook and included mock records of user registrations, login patterns, verification status, and metadata such as browser details. Sensitive fields such as passwords were excluded or masked to maintain privacy standards in the training environment.

Python was used for data extraction, cleaning, and feature engineering. Irrelevant records — such as null entries, placeholder users, or test submissions — were filtered out to improve data quality. The cleaned dataset was exported in CSV format for further analysis using SQL Server and Power BI.

This setup ensured that the dataset was suitable for exploratory data analysis (EDA) while simulating the challenges and complexity of working with real government service data in a secure and ethical manner.

## 3.2 Description of Columns

**🔹 Job Seeker Identity**

* Unique IDs, name, phone, email, gender, date of birth, Aadhaar-related info

**🔹 Login & Security**

* Passwords (initial, current, new), login dates, browser details, login attempts, lock status, security question & answer

**🔹 Registration & Activity**

* Request date, daily serial number, profile submission, satyapan (verification) status, final submission, archival status

**🔹 System & Metadata**

* IP address, insert date, CSC info, eDistrict username, flags (old registration, user registration success, account active/deactivated), internal check flags

**🔹 Social Worker Details (if any)**

* SW unique ID and creation date

**🔹 COVID-related**

* Arogya Setu status

# **Chapter 4: Data Preprocessing**

Data preprocessing is one of the most critical phases in any data analysis or machine learning project. Raw data is often incomplete, inconsistent, or in an unsuitable format for analysis. Therefore, various preprocessing techniques were applied to prepare the data for meaningful analysis and visualization. The objective was to improve data quality, enhance model performance (if applicable), and ensure accurate and consistent insights.

## 4.1 Handling Null Values

The dataset contained several missing values across important columns such as satyapan\_done, Customer\_Rating, js\_first\_log\_date, and LoginAttempts.

**Steps Taken:**

* **Null Identification**: Using df.isnull().sum(), the total missing values in each column were identified.
* **Imputation**:
  + For categorical columns like satyapan\_done, missing values were filled with "Not Provided" or "Unknown" based on context.
  + For numerical columns such as LoginAttempts, 0 was used where a missing value implied no login activity.
* **Drop Null Rows**: In columns like eng\_name\_of\_js or js\_unique\_id—where the information is crucial—rows with missing values were dropped to maintain integrity.

This step ensured that missing values didn’t affect aggregations, groupings, or skew the insights derived from Power BI or Python EDA.

## 4.2 Data Type Conversion

Several columns required **conversion from string/object types** to appropriate data formats to enable mathematical operations and date-based filtering.

**Conversions Included:**

* request\_date, insertdate, js\_first\_log\_date, LastloginDate ➝ Converted to datetime format using pd.to\_datetime().
* is\_verified, old\_reg\_YN, userreg, AadharFlag ➝ Converted to int or boolean for easier filtering.
* phone\_no\_js ➝ Converted to string to avoid truncation or mathematical treatment.

Proper data types helped in sorting, filtering, and calculating durations (e.g., account age, login delays) with precision.

## 4.3 Feature Engineering

To enhance analysis, new **derived features** were created from the existing dataset. These features provided more actionable insights and helped in grouping behaviors for decision-making.

**Key Features Created:**

* login\_status: Binary column indicating whether a user ever logged in or not.
* account\_age\_days: Number of days since the job seeker's data was inserted.
* df['account\_age\_days'] = (pd.to\_datetime('today') - df['insertdate']).dt.days
* verification\_status: Mapped is\_verified and satyapan\_done into descriptive categories.
* login\_frequency\_category: Bucketed LoginAttempts into Low, Medium, High login behavior.

These engineered features helped in clustering users by behavior, identifying inactive accounts, and visualizing platform engagement.

# **Chapter 5: SQL-Based Data Analysis**

SQL (Structured Query Language) was extensively used for conducting initial and intermediate-level data analysis directly within the database environment. By leveraging SQL queries, key performance indicators and patterns were extracted without the need to load the entire dataset into memory. This optimized the workflow and allowed complex aggregations, filters, and data restructuring to be done more efficiently.

# **Chapter 6: Visualization Using Power BI**

Power BI was used as the primary tool for building interactive dashboards and generating insightful visualizations from the processed dataset. With its robust data modeling capabilities and user-friendly interface, Power BI enabled the transformation of raw and structured data into visually rich, interactive reports that supported better understanding, decision-making, and stakeholder communication.

## 6.1 Dashboard Design

The design of the Power BI dashboard was centered on **clarity, usability, and storytelling**. A **modular layout** was followed, with separate pages or sections dedicated to:

* **Login Behaviour**
* **Verification & Conversion**
* **Technical Insights**
* **Growth Over Time**

Each dashboard was structured with a **title bar**, **summary KPIs**, and **visual detail panels**, making it intuitive for both technical and non-technical users.

## 6.2 Filters and Slicers

To make the dashboards interactive and customizable, **Power BI slicers and filters** were incorporated:

* **Date Range Slicers**: Allowed filtering of trends by login dates.
* **Verification Status**: Helped distinguish between verified, partially verified, and unverified users.

These interactive controls allowed users to drill down into specific user segments, observe time-based behavior, and perform ad-hoc comparative analysis.

## 6.3 Key Charts and Metrics

Multiple chart types were used in Power BI to represent different facets of the dataset:

* **Line Charts**: For tracking trends in daily and monthly registrations.
* **Bar and Column Charts**: For comparisons like region-wise user counts, verification rates, and login frequency.
* **Pie/Donut Charts**: To show proportions of profile completion, verification status, or browser usage.
* **Stacked Bars**: To visualize combinations such as verified vs. unverified across regions or zones.
* **KPI Cards**: To display total users, active users, average login attempts, and registration drop-off rates.

## 6.4 User Interaction and Storytelling

Power BI’s interactive features allowed end-users to **explore the data on their own**, helping them derive insights without writing queries or code.

**Features Implemented:**

* **Navigation Buttons**: Used action buttons for seamless movement between dashboard pages.
* **Dynamic Titles and Narratives**: Updated automatically based on selected filters to guide user attention.

This storytelling capability helped stakeholders not only see what was happening in the data but also **understand why**, supporting better decisions and identifying action points.

# Chapter 7. Insights and Actionable Recommendations

This section integrates the key findings from the Power BI dashboards with targeted recommendations to improve user engagement, reduce drop-offs, enhance system usability, and personalize experiences for better platform performance. The insights span user behavior, system usage, and technical operations.

## 7.1 User Engagement

* **93.8%** of users logged in after registration → Strong initial engagement.  
  **→ Recommend:** Post-login prompts, reminder messages, and early engagement nudges.
* Most users logged in within **3–10 days**.  
  **→ Recommend:** Automated reminders within 3 days of inactivity.
* ~8,000 users returned multiple times.  
  **→ Recommend:** Gamification (badges, milestones) and user progress dashboards.

## 7.2 Profile Completion and Verification

* **~44K completed profiles; 47K registered on the same day.  
  → Recommend:** Visual progress bars and completion incentives.
* **6,394 users verified; email verification slightly lower at 5,910.  
  → Recommend:** Simplifying the process and offering CSC support.
* **Funnel shows minimal drop-off post-registration.  
  → Recommend:** Reinforcing with targeted reminders and regional focus.

## 7.3 Technical and Browser Usage

* Major browsers: **Chrome (59K), Firefox (32K),** others.  
  **→Recommend:** Ongoing compatibility testing and UI optimization.
* High use of **older browser versions**.  
  **→ Recommend:** Upgrade prompts and support for legacy systems.
* **No major technical issues** observed.  
  **→ Recommend:** Adding password strength checks

## 7.4 Growth Trends

* **300+ new registrations/day** indicate stable growth.  
  **→ Recommend:** Forecasting infrastructure needs and sharing metrics regionally.

## Data Gaps

* **6.2%** never logged in → Possible drop-offs or test data.  
  **→ Recommend:** Re-engagement campaigns and data review.
* Some **unknown or legacy browsers** recorded.  
  **→ Recommend:** Improving tracking and validation scripts.
* Only **1,596 users** completed all steps.  
  **→ Recommend:** Auditing backend capture and adding auto-save options.
* Email verification slightly lags behind phone.  
  **→ Recommend:** Improving deliverability and in-app prompts.

# **Chapter 8: Conclusion**

This project provided a comprehensive end-to-end analysis of user behavior and platform performance for a large-scale jobseeker registration system. Using a combination of **Python, SQL, and Power BI**, the project effectively transformed raw system data into actionable business insights.

The analysis began with **data collection, preprocessing, and exploration**, covering login behavior, verification flows, profile submissions, and browser preferences. Through Power BI dashboards, the project illuminated critical metrics such as:

* Total registrations, daily averages, and funnel drop-offs
* Login frequency and time-to-login trends
* Profile completion and verification rates
* Regional and technical usage patterns

## Key Accomplishments:

* **93.8% of users successfully logged in** after registration, demonstrating strong initial engagement.
* Nearly **50% of users completed their registration on the same day**, showing smooth onboarding.
* The platform showed **technical reliability across browser versions**, including older ones.
* A **steady growth pattern** in user registration was observed over time, reflecting positive adoption trends.

Further, the project uncovered opportunities for improvement through a series of **strategic recommendations** focused on:

* Boosting ongoing engagement through gamification and notifications
* Reducing drop-offs with simplified flows and assisted registrations
* Enhancing security with 2FA and password policies
* Delivering localized, language-specific, and geo-aware user experiences

## Final Thoughts:

This analysis not only highlighted the strengths of the platform but also presented data-backed strategies to refine and scale its impact. By integrating data insights into operational decision-making, the platform can evolve into a more efficient, inclusive, and user-friendly ecosystem — one that continues to empower jobseekers across diverse regions and backgrounds.

The framework and approach used here can also serve as a blueprint for similar government or enterprise-scale digital platforms seeking to optimize user onboarding, engagement, and retention through data.