# Project Handover Report

Project Name: NPS App (National Polling Survey Application)

Prepared For: NPS  
Prepared By: Development Team  
Date: 03/10/20205

## 1. Project Overview

The NPS App is a survey and voting management system consisting of:

* Backend API (FastAPI) for managing surveys, candidates, and users.
* Admin Management Dashboard for CRUD operations and analytics.
* Mobile App (via API) for survey participation.
* Database (PostgreSQL 14.9) for persistent storage of survey and user data.

This document provides details on server setup, database, and dashboard usage, along with placeholders for credentials and frontend handover.

## 2. Server Information

Primary Server (Production VPS)

* OS: Ubuntu 22.04
* Domain: [https://npsbd.xyz](https://npsbd.xyz/)
* Public IP / Port: 198.23.217.44:5000
* Web Server: Nginx (reverse proxy for FastAPI app)
* Backend Framework: FastAPI (Python 3.10+)
* Frontend Framework: Next.js (Optional – Dashboard UI)

## 3. Database Information

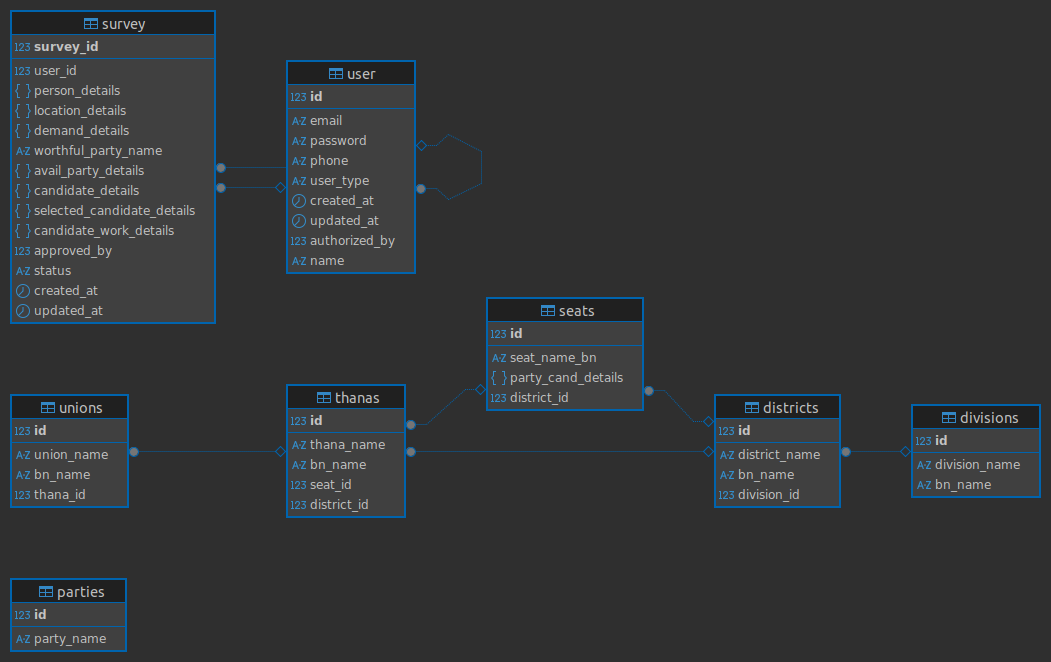
* Database Server: PostgreSQL
* Version: 14.9
* ORM / Query Tool: SQLAlchemy
* Database Name: [To be filled]
* Connection String: [To be filled – format: postgresql://user:password@host:port/dbname]

### 3.1 Key Entities in Database

* Users: Manages admin, super\_admin, and duser roles.
* Surveys: Stores survey metadata, demand details, party/candidate details, responses.
* Candidates/Parties: Stores political parties and candidates.
* Seats / Thanas / Unions / Districts / Divisions: Hierarchical structure for geographical survey mapping.

### 3.2 Database Schema

Below is the relational schema for the project database:



Tables Overview:

1. survey – Core survey data (user\_id, location, party details, candidate info).
2. user – Authentication, roles, authorization.
3. seats – Parliamentary seats, linked to districts.
4. districts – Administrative districts, linked to divisions.
5. divisions – Top-level regional units.
6. thanas – Sub-district level, linked to districts and seats.
7. unions – Lowest administrative units, linked to thanas.
8. parties – Political parties.

## 4. Dashboard Information

* Application Name: NPS Dashboard
* Framework: FastAPI (backend APIs) + Streamlit (optional CRUD UI) + Next.js (planned UI)
* Version: v1.0 (initial production deployment)
* Main Features:
  + Admin & Super Admin access
  + Survey management (CRUD)
  + Candidate and party CRUD operations
  + Analytics endpoints for reporting
  + API docs via Swagger UI (/docs)

## 5. Credentials (To be Filled Separately)

* Server SSH Credentials: [To be filled]
* Database Credentials: [To be filled]
* Dashboard/Admin Credentials: [To be filled]

## 6. Training & Handover

### Database Management Training

* Basic PostgreSQL commands (start/stop, backup, restore).
* Using psql for queries.
* ORM integration through SQLAlchemy.
* Migration handling (if needed).

### Dashboard Usage Training

* Logging in with admin/super\_admin roles.
* Creating, updating, and deleting surveys.
* Viewing analytics reports.
* Managing candidates and political parties.

## 7. Deployment & Services

* Backend API:
  + Runs on FastAPI (with uvicorn / gunicorn).
  + Configured as systemd service (fastapi-app.service).
  + Accessible via: http://npsbd.xyz/api
* Dashboard:
  + Streamlit for CRUD (optional).
  + Next.js frontend (to be finalized).
* Reverse Proxy:
  + Nginx configured to serve FastAPI + static files.

## 8. Appendices

* Repository URL: https://github.com/rafeebubt/nps\_app
* Installation Steps: See README.md (already included in repo).
* Systemd Service File: fastapi-app.service included in repo.
* API Documentation: Swagger docs available at http://npsbd.xyz/api/docs