PROJECT PROPOSAL

1. **Theme –** Artificial Intelligence
2. **Problem Statement:**

   How might we build an AI-powered tool to efficiently analyze petitions, categorize them into relevant departments, flag urgent and important cases, and send reminders to officials, while also identifying repetitive grievances and tracking progress until resolution? The tool should also include a feature to communicate the status of the grievance to the petitioner, ensuring transparency and accountability.
3. **College Code and College Name:**

   2116 – Rajalakshmi Engineering College
4. **Guide Name –** Mr. K Deepak Kumar
   i. **Designation –** Assistant Professor
   ii. **Mobile No. –** 8098758511
   iii. **Email ID –** deepakkumar.k@rajalakshmi.edu.in
5. **Student Team Details**

S.No	**Student Reg. No.**	**Name of the Student**	**Branch**	**Mobile No.**	**Email ID**
1.	2116210701149	MANOJ M G	Computer Science and Engineering	9360270749	210701149@rajalakshmi.edu.in
2.	2116210701158	MITESH A	Computer Science and Engineering	9360438326	210701158@rajalakshmi.edu.in
3.	2116210701180	NITHINPRANAO S	Computer Science and Engineering	9894442981	210701180@rajalakshmi.edu.in

6. Project Summary

This project is an innovative solution designed to revolutionize government petition processing by leveraging advanced AI technologies. The system employs sophisticated natural language processing and machine learning algorithms to automatically categorize petitions, prioritize urgent cases, and route them to appropriate departments. It provides a comprehensive tracking mechanism that monitors petition progress, generates actionable insights on recurring citizen concerns, and ensures transparent communication through personalized status updates. By integrating intelligent classification, real-time tracking, and multi-channel communication infrastructure, the platform aims to enhance administrative efficiency, improve government accountability, and create a more responsive citizen-government interaction model that transforms traditional grievance management processes.

7. Proposed Solution with Methodology

The core architecture will utilize traditional machine learning algorithms for efficient petition processing. The system will employ Support Vector Machines (SVM) and Random Forest classifiers for robust text classification, leveraging scikit-learn's powerful libraries. Text preprocessing will involve techniques like TF-IDF vectorization to transform petition text into meaningful feature representations, enabling accurate department routing and urgency assessment.

The technological stack will comprise Next.js for frontend development, providing a responsive and interactive user interface. MongoDB will serve as the database, enabling flexible document storage and efficient data management. The backend, built with Node.js and Express, will handle petition ingestion, machine learning inference, and communication workflows. A key component will be the classification pipeline that uses SVM and Random Forest models to categorize petitions, predict urgency, and identify recurring themes.

The machine learning workflow will involve creating a labeled dataset from historical petitions, implementing rigorous feature engineering, and training models using a 70-20-10 train-validation-test split. The models will be designed to extract critical information, assess sentiment, and route petitions to appropriate departments with high accuracy. Real-time tracking mechanisms will enable continuous monitoring of petition status, with automated communication channels providing transparent updates to petitioners.

Deployment will follow an incremental strategy, starting with pilot testing and progressively expanding system capabilities. Continuous model performance monitoring and periodic retraining will ensure the platform's effectiveness and adaptability to evolving citizen communication needs. The ultimate goal is to create a streamlined, efficient system that enhances government-citizen interactions through intelligent, data-driven petition management.

8. Workplan / Time Schedule Indicating the Project Mile Stone

Week 1-2: Project Initialization & Data Preparation

- Requirements analysis
- Dataset collection

- Environment setup
- Initial data preprocessing

Week 3-4: Machine Learning Model Development

- SVM and Random Forest model training
- Feature engineering
- Model evaluation and tuning
- Performance benchmarking

Week 5-6: Backend & Infrastructure Development

- API development
- MongoDB integration
- Classification pipeline implementation
- Petition routing logic

Week 7-8: Frontend, Integration & Deployment

- Next.js user interface development
- System integration
- End-to-end testing
- Pilot deployment
- Initial user training

Key Milestones:

1. Functional ML classification system
2. Complete backend infrastructure

3. Operational user interface

4. Successful initial deployment

Project Completion: 8 weeks – 12 weeks

9. Plan of action / Implementation

1. Preparation Phase

- Assemble cross-functional team
- Define detailed technical specifications
- Configure development environment
- Set up version control (GitHub)

2. Data Strategy

- Collect historical petition datasets
- Anonymize and clean data
- Create labeled training dataset
- Define classification categories

3. Machine Learning Development

- Implement text preprocessing pipeline
- Train SVM and Random Forest models
- Develop feature extraction techniques
- Create model evaluation framework

4. Backend Infrastructure

- Design MongoDB schema
- Develop Node.js/Express APIs
- Implement classification microservices
- Create secure authentication mechanism

5. Frontend Development

- Design responsive Next.js interface
- Develop dashboard components
- Implement real-time status tracking
- Create user notification system

6. Integration & Testing

- Combine ML models with backend
- Perform comprehensive system testing
- Conduct performance benchmarking
- Security vulnerability assessment

7. Deployment Preparation

- Configure cloud hosting
- Set up continuous integration
- Develop monitoring dashboards
- Create user documentation

8. Pilot Launch

- Controlled initial deployment

- Gather user feedback
- Monitor system performance
- Iterative improvements

10. List of facilities available in college

PC Systems, Internet

11. Nature of Industry support

Not required (Maybe some support from the organization that will use this website)

12. Total Cost

Monthly Cloud & Database Resources Cost Estimate:

MongoDB Atlas:

- Shared Tier: Free

- Dedicated Small Cluster: ₹5,000-7,000/month

AWS Cloud Services:

- Small EC2 Instance: ₹3,500-5,000/month

- Elastic Beanstalk Deployment: ₹2,000-3,500/month

- Data Transfer & API Gateway: ₹1,500-2,500/month

Total Yearly Estimated Cost: ₹1,44,000-2,16,000

Note - Cost may vary depending on usage

13. Details of Financial Assistance Required

No assistance required during development

14. Expected Results / Outcome

The AI-Powered Petition Management Platform is expected to deliver transformative outcomes, including 40-50% reduction in administrative processing time, enhanced government accountability through transparent tracking, and data-driven insights into citizen grievances. The system will enable intelligent petition routing, prioritize urgent cases, provide real-time status communication, and generate comprehensive trend reports that help identify systemic issues, ultimately improving citizen-government interactions by creating a more responsive, efficient, and transparent grievance management ecosystem.