# 🛖 ThunderByte 2025 Hackathon

**Theme**: Tech for a Rainy Day – Building Solutions for the Monsoon Season

**Duration**: One Day

**Format:** 4 Person in the group

Streams:

March Person
 Full Stack Development

III Data Science

## **Important Dates**

**Event Date:** 

Venue:

Registration Start: Registration Ends: Registration Link:

## FULL STACK DEVELOPMENT – Problem Statements

## 1. Rain Alert & Safety App

**Problem Statement**: Design a web app that provides real-time monsoon weather alerts, emergency contacts, and community updates.

### Features:

- User registration/login
- Live weather updates using API (e.g., OpenWeatherMap)
- Geo-based alerts for flood-prone areas
- Section for emergency contacts (hospitals, shelters, helplines)
- Community feed for users to post real-time updates (optional moderation)

## Expected Deliverables:

- Fully responsive web app
- Integration with weather API
- GitHub repo with:
  - Code (frontend/backend)
  - Setup instructions
  - o API usage
- Deployed version (on Netlify/Vercel/Render/Heroku)

### 2. Local Travel Assistant for Monsoon

**Problem Statement**: Build a monsoon-friendly travel assistant web app that helps users plan safe and dry travel within the city.

#### Features:

- Route planning avoiding waterlogged/flooded roads
- Crowd-sourced reporting of road conditions
- Suggested alternate safe routes
- Transport availability checker (mock or API integration)

#### Expected Deliverables:

- Frontend + backend integration
- Interactive map UI (Leaflet.js / Google Maps API)
- Real-time route updates (mock data allowed)
- GitHub repo with clear documentation
- Deployable web version

## 3. Monsoon Recipe Hub

**Problem Statement**: Create a cozy recipe-sharing platform where users can browse and upload monsoon-special recipes.

#### Features:

- User authentication
- Recipe submission with photo and tags (e.g., "spicy", "chai", "snacks")
- Search and filter by type or ingredient
- Like/comment functionality
- Responsive design

#### Expected Deliverables:

- React frontend
- Node/Spring Boot/Django backend
- MongoDB/MySQL for DB
- GitHub repo with code and instructions
- Deployed project link

## DATA SCIENCE – Problem Statements

### 1. Rainfall Pattern Prediction

Problem Statement: Build a model to predict rainfall in Indian cities using historical rainfall data.

#### Tasks:

- Load and clean public rainfall datasets (e.g., IMD data)
- Build regression or time-series models (e.g., LSTM, ARIMA)
- Visualize seasonal patterns

• Predict rainfall for next 7 days in selected cities

#### Expected Deliverables:

- Jupyter Notebook or Python script
- Model file (joblib/pkl)
- Visualizations (matplotlib/seaborn/Plotly)
- README with approach and model evaluation
- GitHub repo with organized code and data

## 2. Monsoon Tweet Sentiment Analyzer

**Problem Statement**: Analyze public sentiment during monsoon using Twitter data.

#### 🧠 Tasks:

- Collect tweets using Tweepy (filter for keywords like "rain", "flood", "monsoon")
- Clean text (stopwords, lemmatization)
- Perform sentiment analysis (TextBlob/VADER/transformers/Other)
- Generate dashboards with most frequent words, sentiment trends, geo-tags if available

### Expected Deliverables:

- Sentiment analysis notebook
- Wordclouds, pie charts, sentiment distribution graphs
- Dashboard (optional: Streamlit app)
- GitHub repo with code and results
- README.MD file

#### 3. Flood Risk Classifier

Problem Statement: Predict flood risk in districts based on weather and geographic features.

#### Tasks:

- Use datasets combining rainfall, elevation, river proximity, etc.
- Clean and merge data
- Build classification model (Logistic Regression, RF, XGBoost)
- Evaluate with metrics (accuracy, precision, recall)
- Identify top contributing features

#### Expected Deliverables:

- Jupyter notebook with EDA, modeling, evaluation
- Data visualization of high-risk areas
- Model file (optional)
- GitHub repo with code and data
- README explaining methodology

## General Hackathon Flow

Time	Activity	
9:00 AM	Registrations + Opening Ceremony	
9:30 AM	Problem Statement Briefing & Team Formation	
10:00 AM	Hacking Begins	
1:00 PM	Check-in & Mid-Day Mentorship	
5:00 PM	Code Freeze + Submission on GitHub	
6:00 PM	Project Demos	
7:00 PM	Judging & Results	
7:30 PM	Closing Ceremony	

## 📌 Submission Format

Each team must submit the following on GitHub:

- Codebase (cleaned and structured)
- README file with:
  - o Project title and problem statement
  - $\circ \quad \text{Team members and tech stack} \\$
  - Setup instructions
  - Screenshots or video (if possible)
- Deployed link (if applicable)
- Dataset (for DS projects)

## Evaluation Criteria – Total: 100 Points

Category	Description Points	
1. Technical Implementation	Code quality, use of appropriate tools/libraries, working functionality, and integration of APIs or models.	30 Points
2. Innovation & Creativity	Uniqueness of the idea, originality in solving the problem, creative features or user experience.	25 Points
3. Relevance to Problem Statement	How well the solution addresses the given problem statement and aligns with the monsoon theme.  25 Poin	
4. Presentation & Deliverables	Clarity of GitHub README, UI/UX, working demo (if applicable), documentation, and overall presentation.	20 Points

## 💃 Awards Breakdown (per stream)

Position	🏆 Award	
1st Place	Certificate	Gold Medal 🥇
2nd Place	Certificate	Silver Medal 🥈
3rd Place	Certificate	Bronze Medal 🥉