Project READMEs

# MERN-stack-Project

# Adding Places This is a web application that allows users to register, log in, and manage their personalized places. Each place includes a title, image, and location on a map. Users can view, edit, or delete the places they’ve added.

# Features User authentication (register and login) Add new places with details and map location View all user-specific places Edit or delete existing places Google Maps integration for location selection and display Tech Stack Frontend React Backend Node, Express Database MongoDB Map Google Maps API

# Prerequisites Node.js and npm installed MongoDB running locally or connection string for remote DB Google Maps API key How to Step 1. Clone the repository

# Step 2. Open a terminal at the root directory of the project

# Step 3. Run 'cd backend' at the terminal

# Step 4. Run 'npm install' at the terminal

# Step 5. Run 'npm run server' at the terminal

# Step 6. Run 'cd frontend' at the terminal

# Step 7. Run 'npm install' at the terminal

# Step 8. Run 'npm start' at the terminal

# Step 9. Open [http://localhost:3000]

# Credits Built using MERN and Google Maps

# Python-Package-app20

## Python-Package-app20  
  
This repository contains a simple Python package related to invoice processing.  
  
### Features:  
- Folder structure includes `invoices/`, `invoicing/`, and a `main.py` script  
- Likely processes or generates invoices as packages  
- Suitable for packaging practice or invoicing automation  
  
---

# Threaded Email and Image Cleaner App

## Threaded Email and Image Cleaner App  
  
This Python project uses threading to detect motion through a webcam and send an email with a captured image once movement stops. It leverages OpenCV and multithreading to capture frames and send notifications.  
  
### Key Features:  
- Motion detection using OpenCV  
- Image saving on detection  
- Email notification when motion stops  
- Automatic image cleanup using `atexit`  
- Thread-based concurrency for responsiveness  
  
---

# Book Analysis with NLP

## Book Analysis with NLP  
  
This project performs Natural Language Processing (NLP) tasks on the book "Miracle in the Andes".  
  
### Key Features:  
- Loads and parses a text file  
- Counts chapters using string methods and regex  
- Identifies frequency of the word "love"  
- Builds a word frequency distribution excluding stopwords  
- Performs sentiment analysis on each chapter using VADER  
  
---

# Weather Forecast Data App

## Weather Forecast Data App  
  
This is a Streamlit web app that displays weather forecasts for user-specified locations using OpenWeatherMap API.  
  
### Key Features:  
- User input for location and number of forecast days  
- Displays temperature trends via Plotly line charts  
- Shows sky condition icons (clear, rain, snow, etc.)  
- Streamlit UI elements: text input, slider, dropdown  
  
---

# Weather-API (Flask)

## Weather-API (Flask)  
  
This Flask-based web app serves weather data from CSV files using API endpoints. It reads temperature data and serves it by station ID, date, or year.  
  
### Key Features:  
- Flask routes for: specific day, full history, and yearly data  
- Uses Pandas for CSV handling and filtering  
- Minimal web interface via Jinja templates  
  
---

# Invoice Generator

## Invoice Generator  
  
This project reads Excel files and generates invoice PDFs from them.  
  
### Key Features:  
- Reads multiple Excel invoice files  
- Extracts invoice data like customer, date, products, and prices  
- Generates formatted PDF invoices with headers and totals  
- Outputs to a `PDFs/` directory  
  
---

# My Portfolio Web App

## My Portfolio Web App  
  
This Python web app is a personal portfolio site that displays Python projects and allows sending emails through a contact form.  
  
### Key Features:  
- Displays data from CSV file (e.g. projects, descriptions)  
- Contact form that sends an email  
- Organized using folders like `images/`, `pages/`  
  
---

# Scraping Tours SQL App

## Scraping Tours SQL App  
  
This app scrapes tour listings from a webpage, checks for duplicates in an SQLite database, and sends email notifications for new events.  
  
### Key Features:  
- Web scraping using `requests` and `selectorlib`  
- SQLite integration to track past entries  
- Sends email alerts for new events  
- Object-oriented structure with `Event`, `Database`, `Email` classes  
  
---

# Restaurant Menu App (Django)

## Restaurant Menu App (Django)  
  
A Django-based web app to manage a restaurant's menu using a database and admin interface.  
  
### Key Features:  
- Menu item management with CRUD operations  
- Django admin panel for easy entry management  
- QR code generation for accessing menu items  
- Uses Bootstrap and Jinja templates for styling  
  
---

# Hotel Booking App

## Hotel Booking App  
  
This project allows checking hotel availability and making reservations.  
  
### Key Features:  
- Uses CSV files to manage hotels and availability  
- Class-based structure with `Hotel` and `ReservationTicket`  
- Simple object-oriented implementation of booking logic  
  
---

# Flask Form App

## Flask Form App  
  
This is a Flask web application that collects user form submissions and sends email confirmations.  
  
### Key Features:  
- Form fields: name, email, date, occupation  
- Stores data in SQLite via SQLAlchemy  
- Sends confirmation emails using Flask-Mail  
- Flash messages for success/failure  
  
---

# Django Form App

## Django Form App  
  
A Django-based application for job applications that saves user data via Django forms and admin.  
  
### Key Features:  
- Django project setup with models  
- Form submissions are stored in SQLite  
- Admin interface available  
  
---

# App4 PDF Generator

## App4 PDF Generator  
  
This app generates a PDF from a `topics.csv` file by creating pages for each topic and appending footers.  
  
### Key Features:  
- Reads topic names from CSV  
- Creates and merges PDFs with footer text  
- Saves combined output to `output.pdf`  
  
---

# To-Do App with GUI and Web Support

## To-Do App with GUI and Web Support  
  
This project implements a To-Do application with multiple interfaces: CLI, GUI using FreeSimpleGUI, and web using Streamlit.  
  
### Key Features:  
- GUI with checklist, add/edit/delete options  
- Command-line support with `cli.py`  
- Streamlit-based web version  
- Stores tasks in `todos.txt`  
  
---

# Attack Detection in Collaborative Filtering Recommender System

## Attack Detection in Collaborative Filtering Recommender System  
  
A B.Tech project focused on detecting profile injection (shilling) attacks in collaborative filtering recommender systems.  
  
### Key Highlights:  
- Discusses push and nuke attacks  
- Literature review of methods (e.g., Hidden Markov Models, clustering, hybrid filters)  
- Covers motivation, methodologies, and impact on system reliability  
  
---

# Music House App

## Music House  
  
A full-stack music streaming and admin management app.  
  
### Features:  
- Users can listen, favorite, and manage songs  
- Admins can manage users, albums, songs, and artist data  
- Integrated with Firebase and MongoDB  
  
### Stack:  
- Frontend: ReactJS, HTML, CSS  
- Backend: NodeJS, Express  
- Hosting: Netlify (frontend), Heroku (backend)  
  
---

# Stock Market Prediction Using LSTM and Monte Carlo Simulation

## Stock Market Prediction Using LSTM and Monte Carlo Simulation  
  
This repository presents two approaches for predicting stock market prices: LSTM and Monte Carlo Simulation using historical AAPL data.  
  
### Features:  
- LSTM for time-series trend prediction  
- Monte Carlo for probabilistic risk modeling  
- Evaluation using RMSE, rolling volatility analysis, scenario projections  
  
---