**import 'dotenv/config';** This imports the dotenv package to load environment variables from a .env file into process.env. It allows you to manage configuration settings, like port numbers, without hardcoding them in your application.

**Handling Uncaught Exceptions:** process.on('uncaughtException', (err) => { });This event listener catches exceptions that weren't handled elsewhere in the application.

**Handling Unhandled Promise Rejections:** process.on('unhandledRejection', (err) => {});This event listener handles promise rejections that weren't caught.

**Graceful Shutdown on SIGTERM:** process.on('SIGTERM', () => { }); The listener for the SIGTERM signal (often sent by process managers or when you manually terminate the app) logs a message and closes the server gracefully, ensuring that ongoing requests can finish before the process exits.

process.on is a method in Node.js that allows you to listen for specific events that occur in the process.

 **@babel/polyfill**: Provides support for new JavaScript features in older environments by emulating missing functionality.

 **axios**: A promise-based HTTP client for making requests to servers (e.g., GET, POST) in Node.js and the browser.

 **bcryptjs**: A library for hashing passwords using the bcrypt algorithm for secure storage.

 **body-parser**: Middleware for parsing incoming request bodies in JSON or URL-encoded formats.

 **casual**: A library to generate random data for testing or seeding databases.

 **compression**: Middleware that compresses HTTP responses to reduce payload size and improve performance.

 **cookie-parser**: Middleware that parses cookies from the HTTP request header.

 **cors**: Middleware that enables Cross-Origin Resource Sharing (CORS), allowing servers to handle requests from different domains.

 **dotenv**: Loads environment variables from a .env file into process.env to manage configuration.

 **express**: A web framework for building RESTful APIs and web applications in Node.js.

 **express-rate-limit**: Middleware that limits repeated requests from the same IP to protect against DoS attacks.

 **express-session**: Middleware for managing user sessions and storing session data on the server.

 **helmet**: Middleware that adds security headers to protect Express apps from common vulnerabilities.

 **hpp**: Middleware to protect against HTTP Parameter Pollution (HPP) attacks.

 **html-to-text**: Converts HTML content to plain text, often used for email or other plain text processing.

 **jsonwebtoken**: A library for generating and verifying JSON Web Tokens (JWT) for authentication and authorization.

 **knex**: A SQL query builder for Node.js that supports various database engines, like MySQL and PostgreSQL.

 **morgan**: Middleware that logs HTTP requests in the console for easier debugging and monitoring.

 **multer**: Middleware for handling file uploads in Express applications.

 **mysql2**: A MySQL client for Node.js that provides promises-based and performant database querying.

 **nodemailer**: A library for sending emails from Node.js applications.

 **nodemon**: A development tool that automatically restarts the server when files change, improving the development workflow.

 **passport**: Authentication middleware for Node.js that integrates various authentication strategies.

 **passport-github2**: A GitHub authentication strategy for Passport to allow GitHub OAuth login.

 **pug**: A template engine for Node.js that helps generate HTML with a simplified syntax.

 **sharp**: A high-performance image processing library, commonly used for resizing and converting images.

 **slugify**: Converts strings into URL-friendly slugs, typically used for SEO-friendly URLs.

 **stripe**: A Node.js SDK for integrating Stripe's payment processing services into applications.

 **uniqid**: A library to generate unique IDs, often used for identifying resources like database records.

 **validator**: A library for validating and sanitizing strings (e.g., emails, URLs, etc.).

 **xss-clean**: Middleware to prevent Cross-Site Scripting (XSS) attacks by sanitizing input.