# **Performance Testing Template**

**Model Performance Test**

|  |  |
| --- | --- |
| Date | 16 July 2025 |
| Team ID | LTVIP2025TMID59690 |
| Project Name | HouseHunt – Rental Property Recommender |
| Maximum Marks | 2 Marks |

## **Express.js (Node.js):**

**1. Introduction to Express.js:**

Express.js is a minimal and flexible Node.js web application framework that provides a robust set of features for web and mobile applications. It simplifies the process of building server-side logic and APIs using JavaScript.

It is widely used in full-stack applications, especially in the MERN stack (MongoDB, Express, React, Node.js) architecture.

**2. Why Express.js Was Used:**

Express.js was chosen for the project due to the following reasons:

* It enables rapid development of RESTful APIs.
* Simplifies routing, middleware, and request handling in the backend.
* Fully compatible with **Node.js**, enabling server-side development using JavaScript.
* Integrates easily with MongoDB using **Mongoose**.
* Provides middleware support for authentication (e.g., JWT), CORS, file uploads, etc.

**3. How Express.js Was Used in the Project**

In this MERN full-stack application, Express.js was used to:

* Define RESTful API endpoints for **User**, **Property**, and **Admin** management.
* Handle HTTP requests and responses between frontend and database.
* Connect to MongoDB using **Mongoose**.
* Use **JWT Authentication** middleware to protect routes.
* Implement routing modules for Owners, Admins, and Renters.
* Perform CRUD operations on collections like users, properties, bookings, etc.

**4. Sample Route Snippet:**

const express = require('express');

const router = express.Router();

const propertyController = require('../controllers/propertyController');

// Create a new property listing

router.post('/create', propertyController.createProperty);

// Get all properties

router.get('/list', propertyController.getAllProperties);

module.exports = router;

**5. Conclusion:**

Express.js played a vital role in developing a scalable and maintainable backend for the MERN application. Its simplicity, speed, and flexibility allowed the development team to build a robust RESTful API system and integrate it seamlessly with MongoDB and React**.**