# Quick_type Project Report

## Executive Summary

This report details the **Quick_type** web application, a MERN stack-based typing test platform designed to help users improve their typing skills. The system provides real-time feedback, performance tracking, and a secure environment for users to enhance their typing proficiency.

## Table of Contents

## 1. Introduction

In today's digital landscape, typing proficiency is a critical skill for both students and professionals. **Quick_type** addresses this need by providing an intuitive web application that allows users to measure and improve their typing speed and accuracy. Built using the MERN (MongoDB, Express.js, React, Node.js) stack, the application delivers a seamless experience with real-time performance metrics and personalized tracking.

## 2. Problem Statement & Solution

### Challenges Addressed:

- Lack of effective tools to monitor typing performance
- Absence of real-time feedback mechanisms
- Limited data security in existing solutions
- Minimal engagement in traditional typing tools

### Our Solution:

**Quick_type** provides a comprehensive platform that combines accurate performance tracking with an engaging user experience. The system implements secure user authentication via JWT, stores encrypted user data using bcrypt, and delivers immediate feedback on typing tests through a responsive interface.

## 3. Key Features

- **Real-time Assessment**: Instant calculation and display of WPM (Words Per Minute) and accuracy

- **Secure Authentication**: JWT-based user sessions with encrypted password storage

- **User Profiles**: Personalized accounts storing typing history and performance analytics

- **Responsive Design**: Fully adaptive interface optimized for both desktop and mobile devices

- **Performance Analytics**: Comprehensive metrics tracking progress over time

- **RESTful Architecture**: Well-structured API endpoints for seamless frontend-backend communication

## 4. Technical Implementation

**Quick_type** leverages modern web technologies to deliver a robust and scalable solution:

- **Frontend**: React.js with Tailwind CSS for a responsive and aesthetically pleasing UI

- **Backend**: Node.js and Express.js creating a robust application server

- **Database**: MongoDB for flexible and scalable data storage

- **Authentication**: JWT implementation with bcrypt password hashing

- **API Design**: RESTful endpoints for user management and typing test operations

## 5. Project Outcomes

The **Quick_type** project has successfully delivered:

- A fully functional typing assessment platform

- Secure user management system with encrypted data storage

- Comprehensive performance tracking with historical data

- Mobile-responsive interface accessible across devices

- Scalable backend architecture supporting future enhancements

Future roadmap includes competitive features, achievement systems, and expanded test categories.

## 6. Code Architecture

**Authentication Middleware (JWT Implementation)**

```javascript
const jwt = require("jsonwebtoken");

module.exports = function (req, res, next) {
  const token = req.header("Authorization");
  if (!token) return res.status(401).send("Access Denied");

  try {
    const verified = jwt.verify(token, process.env.TOKEN_SECRET);
    req.user = verified;
    next();
  } catch (err) {
    res.status(400).send("Invalid Token");
  }
};
```

## Typing Test Data Model

```javascript
const mongoose = require("mongoose");

const TypingTestSchema = new mongoose.Schema({
  user: {
    type: mongoose.Schema.Types.ObjectId,
    ref: "User",
    required: true
  },
  wpm: {
    type: Number,
    required: true
  },
  accuracy: {
    type: Number,
    required: true
  },
  createdAt: {
    type: Date,
    default: Date.now
  }
});

module.exports = mongoose.model("TypingTest", TypingTestSchema);
```

## User Model with Password Encryption

```javascript
const mongoose = require("mongoose");
const bcrypt = require("bcrypt");

const UserSchema = new mongoose.Schema({
  username: {
    type: String,
    required: true,
    unique: true
  },
  email: {
    type: String,
    required: true,
    unique: true
  },
  password: {
    type: String,
    required: true
  },
  joinDate: {
    type: Date,
    default: Date.now
  }
});

// Pre-save middleware to hash passwords
UserSchema.pre("save", async function(next) {
  if (!this.isModified("password")) return next();

  try {
    const salt = await bcrypt.genSalt(10);
    this.password = await bcrypt.hash(this.password, salt);
    next();
  } catch (error) {
    next(error);
  }
});

module.exports = mongoose.model("User", UserSchema);
```

## 7. Conclusion

**Quick_type** demonstrates the effective application of modern web technologies to create an educational and performance-oriented application. The project balances technical sophistication with user-centered design, providing a valuable tool for typing skill development. Through its implementation of secure

authentication, real-time feedback systems, and comprehensive performance tracking, **Quick_type** offers a complete solution for users seeking to improve their typing proficiency.

## 8. References

- React.js: https://react.dev
- Node.js: https://nodejs.org/en
- Express.js: https://expressjs.com
- MongoDB: https://mongodb.com
- Tailwind CSS: https://tailwindcss.com
- JSON Web Tokens: https://jwt.io
- Bcrypt: https://www.npmjs.com/package/bcrypt

---

**Project Team:**

- Prajwal (2210992057)
- Prince Thakur (2210992092)
- Ishaan (2210991684)

**Submitted To:** Mr. Rahul Sir

**Institution:** Chitkara University Institute of Engineering and Technology, Punjab