RANI CHANNAMMA UNIVERSITY, BELAGAVI.
2024 – 2025

Internship Report
ON
Mern Stack Development

An internship is submitted in partial fulfillment of the requirement for the award of the degree of

BACHELOR OF COMPUTER APPLICATIONS

Submitted by

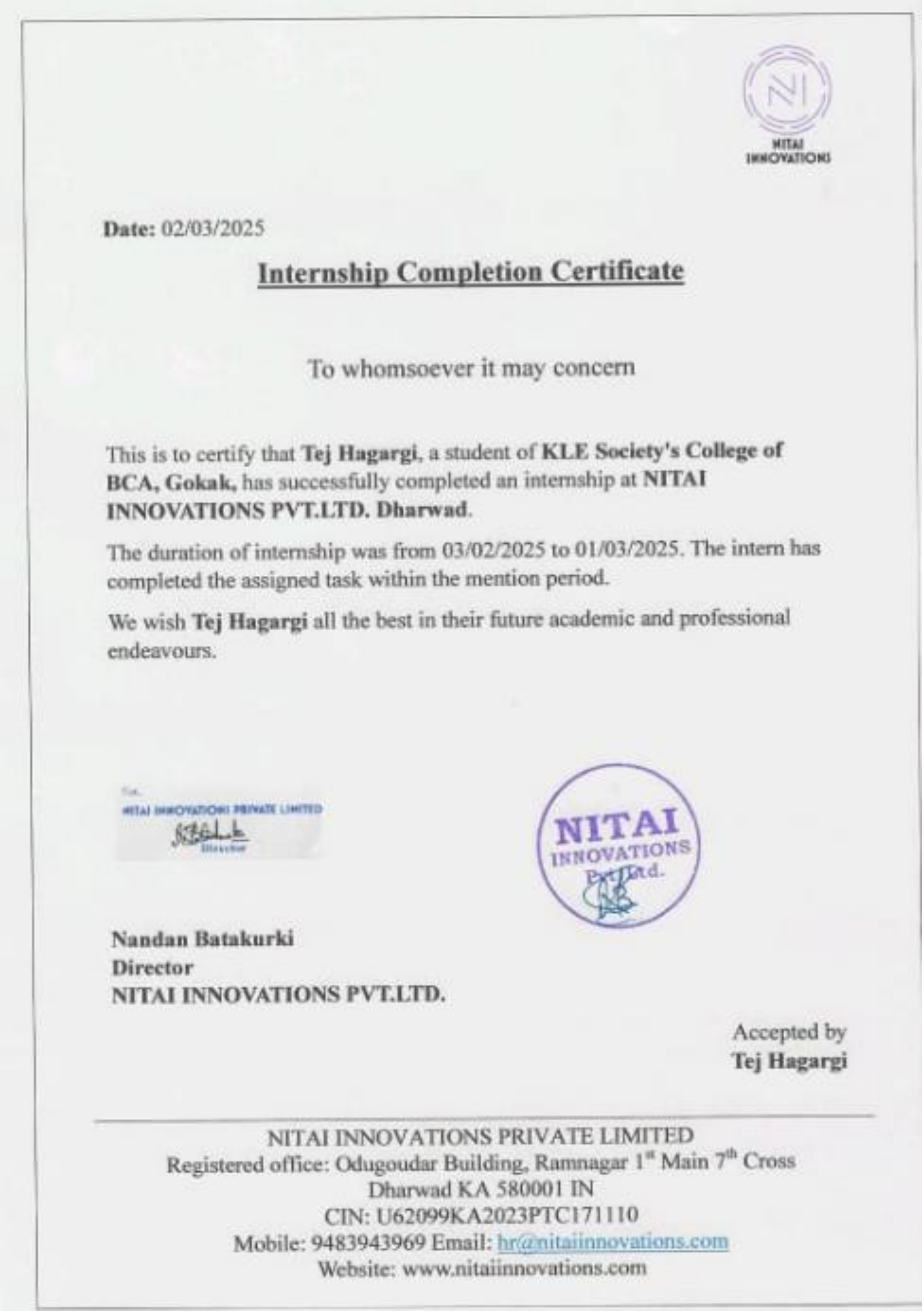
Tej Hagargi [U15EW22S0032]

Under the Guidance of

Prof. Akshata Pethe and Nitai Innovations Private Limited

COLLEGE OF BACHELOR OF COMPUTER APPLICATION GOKAK

2024 - 25



NITAI INNOVATIONS

Date: 02/03/2025

Internship Completion Certificate

To whomsoever it may concern

This is to certify that **Tej Hagargi**, a student of **KLE Society's College of BCA, Gokak,** has successfully completed an internship at **NITAI INNOVATIONS PVT.LTD. Dharwad**.

The duration of internship was from 03/02/2025 to 01/03/2025. The intern has completed the assigned task within the mention period.

We wish **Tej Hagargi** all the best in their future academic and professional endeavours.

For, NITAI INNOVATIONS PRIVATE LIMITED
Director

NITAI INNOVATIONS Pvt.Ltd.

Nandan Batakurki
Director
NITAI INNOVATIONS PVT.LTD.

Accepted by
Tej Hagargi

NITAI INNOVATIONS PRIVATE LIMITED
Registered office: Odugoudar Building, Ramnagar 1st Main 7th Cross
Dharwad KA 580001 IN
CIN: U62099KA2023PTC171110
Mobile: 9483943969 Email: hr@nitaiinnovations.com
Website: www.nitaiinnovations.com

KLE Society's

College of Bachelor of Computer Application (BCA)

Gokak

(Affiliated to Rani Channamma University, Belagavi, AICTE Approved Recognised by-Govt of Karnataka)
Near Byali Kata Gokak-591307 (Dist-Belgavi) Email- klebcagokak@gmail.com

NOC FOR INTERNSHIP

Sub:- No Objection Certificate for Internship Programme at________________.

It is certified that Mr/Ms.____ is a bonafide student with Reg No.__________________ of 6th semester BCA of our college KLE Society's College of BCA Gokak.

The KLE Society's College of BCA Gokak has no objection for doing the Internship programme at ____ for the period from ____________to___________. It is also certified that he/she is not registered for any course requiring, his/her attendance in the class during said this period.

The conduct of the student as recorded by the KLE Society's College of BCA Gokak has been found good.

Date: 09/05/2025
Place: Gokak

KLES's BCA Institute • Gokak

Principal
KLE Society's College of BCA,
Gokak.

TABLE OF CONTENTS

1	COMPANY PROFILE
2	ON JOB TRANING – 1
3	ON JOB TRANING – 2
4	USE CASE 1 AND USE CASE 2

COMPANY PROFILE

This report presents a comprehensive overview of my three-month internship at Nitai Innovations Private Limited, a rapidly growing technology company headquartered in Dharwad Karnataka. Nitai Innovations is renowned for its expertise in software development, cybersecurity, artificial intelligence (AI), and machine learning (ML), catering to a diverse range of industries. The company's mission to deliver secure, innovative, and scalable technological

solutions has provided an enriching and intellectually stimulating environment for skill development and professional growth.

Company Name: Nitai Innovations Pvt. Ltd.

Location: Dharwad, Karnataka, India

Year of Establishment: 2023

Domain Expertise: Software Development, Artificial Intelligence (AI), Machine Learning

(ML), Cybersecurity

Founders:

- Nitin Odugoudar,
- Nandan V Batakurki,

About the Company:

Nitai Innovations Private Limited (NIPL) is a Private Limited Indian Non-Government

Company incorporated in India on 12 March 2023 (Two years and two months 2 days old). Its registered office is in Dharwad, Karnataka, India.

ON JOB TRAINING - 1

Setting up a **MERN stack development environment** involves preparing your system to develop applications using **MongoDB, Express.js, React, and Node.js**. Here's a step-by-step guide to get started:

Step 1: Install Node.js and npm

Node.js is the JavaScript runtime used for both backend development and for running React's development tools.

- **Download and install Node.js** from: https://nodejs.org
- Verify installation:
- node -v
- npm -v

Step 2: Set Up MongoDB

MongoDB is the NoSQL database used in the MERN stack.

Option 1: Install MongoDB locally

- Download MongoDB: https://www.mongodb.com/try/download/community
- Start MongoDB service:
- mongod

Option 2: Use MongoDB Atlas (cloud)

- Create an account at https://www.mongodb.com/cloud/atlas
- Create a new cluster and database
- Get the connection string (used later in backend code)

Step 3: Initialize the Backend (Node + Express)

- Create a backend folder:

- mkdir backend
- cd backend
- npm init -y
- Install dependencies:
- npm install express mongoose cors dotenv
- npm install --save-dev nodemon
- Create the core files:
    - index.js or server.js: entry point
    - .env: to store environment variables
    - Create folders: /routes, /models, /controllers
- Add a script to package.json:
- "scripts": {
-   "start": "node index.js",
-   "dev": "nodemon index.js"
- }

Step 4: Setup React Frontend

- Go to the root directory and create the frontend:
- npx create-react-app frontend
- cd frontend
- npm start

Step 5: Connect Frontend with Backend

- Enable CORS in backend:
- const cors = require("cors");
- app.use(cors());

- In React, make HTTP requests to your backend using fetch or libraries like axios:
- npm install axios
- Use environment variables for API URLs in React:
  Create a .env file in /frontend:
- REACT_APP_API_URL=http://localhost:5000

Step 6: Run the Development Servers

- Run backend:
- cd backend
- npm run dev
- Run frontend:
- cd frontend
- npm start

ON JOB TRAINING – 2

Project Title: Contact Saver

Project Overview

This is a basic MERN stack project where users can create, view, edit, and delete contact information. Each contact includes a name, phone number, and email. It's ideal for beginners and can be completed comfortably within a 1-month internship.

Step-by-Step Setup

Step 1: Install Node.js and MongoDB

- Download Node.js from nodejs.org
- Use MongoDB locally or create a cluster on MongoDB Atlas

Step 2: Backend Setup

- Initialize Node.js project with npm init -y
- Install packages:
- npm install express mongoose cors dotenv
- npm install --save-dev nodemon
- Set up a basic Express server with:
    - A connection to MongoDB
    - RESTful API routes for:

    - Adding a contact (POST)
    - Viewing all contacts (GET)
    - Editing a contact (PUT)
    - Deleting a contact (DELETE)

Step 3: Frontend Setup

- Create a React app using:
- npx create-react-app frontend
- Install Axios:
- npm install axios
- Create a form to add/update contact details and a list to show all saved contacts.
- Use Axios to make API requests to the backend:
  - GET to fetch contacts
  - POST to add contacts
  - PUT to update contacts
  - DELETE to remove contacts

🔷 Step 4: Connect Frontend to Backend

- Add CORS middleware in the backend to allow frontend access.
- Set a proxy in package.json of React to point to backend:

- "proxy": "http://localhost:5000"

Step 5: Run the App

- Start backend:
- npm run dev
- Start frontend:
- npm start

Final Features

- Add a new contact
- View all contacts
- Edit existing contact details
- Delete a contact

USE CASE 1 AND USE CASE 2

Here are **Use Case 1** and **Use Case 2** for the **Contact Saver** project:

Use Case 1: Add New Contact

Use Case Name: Add Contact
Primary Actor: User
Preconditions:

- The user is on the main page of the application.
- The backend server is running and connected to MongoDB.

Main Flow:

1. The user fills out the form with contact details: name, phone number, and email.
2. The user clicks the "Add Contact" button.
3. The frontend sends a POST request to the backend with the contact data.
4. The backend validates the data and saves it to the MongoDB database.
5. The backend returns a success response.
6. The frontend updates the contact list to include the new contact.

Postconditions:

- The new contact is saved and displayed in the contact list.

Use Case 2: Edit Existing Contact

Use Case Name: Edit Contact
Primary Actor: User
Preconditions:

- The user sees the list of saved contacts.
- The user selects a contact to edit.

Main Flow:

1. The user clicks the "Edit" button next to a contact.
2. The selected contact's details are populated into the form.
3. The user modifies the information and clicks the "Update Contact" button.
4. The frontend sends a PUT request to the backend with the updated data and contact ID.
5. The backend finds the contact in the database and updates its record.
6. The backend returns a success response.
7. The frontend updates the displayed list with the new contact details.

Postconditions:

- The selected contact’s information is updated in both the frontend and the database.