



**VIVEKANAND  
EDUCATION SOCIETY**  
**INSTITUTE OF TECHNOLOGY  
(AUTONOMOUS)**

(Affiliated to University of Mumbai, Approved by AICTE & Recognized by Govt. of Maharashtra)

**Project Report of:  
Finance Tracker**

**Full Stack Development (FSD)**  
in Semester - V by

**Atharva Chaudhari (D15A/62)**

**Sameer Jagiasi (D15A/64)**

under the guidance of

**Mrs. Pooja Prajapati**



**Department of Information Technology**

**Vivekanand Education Society's Institute of Technology-2025-2026**



# **VIVEKANAND**

## **EDUCATION SOCIETY**

---

### **INSTITUTE OF TECHNOLOGY**

#### **(AUTONOMOUS)**

(Affiliated to University of Mumbai, Approved by AICTE & Recognized by Govt. of Maharashtra)

<b>SR NO</b>	<b>TABLE OF CONTENTS</b>	<b>PAGE</b>
<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Literature Review</b>	<b>2</b>
<b>3</b>	<b>Resources Required</b>	<b>2</b>
<b>4</b>	<b>Flow of Making Project</b>	<b>3</b>
<b>5</b>	<b>OUTPUT</b>	<b>8</b>
<b>6</b>	<b>Challenges &amp; Solutions</b>	<b>9</b>
<b>7</b>	<b>Future Enhancements</b>	<b>9</b>
<b>8</b>	<b>Conclusion</b>	<b>10</b>
<b>9</b>	<b>References</b>	<b>10</b>



# VIVEKANAND EDUCATION SOCIETY INSTITUTE OF TECHNOLOGY (AUTONOMOUS)

(Affiliated to University of Mumbai, Approved by AICTE & Recognized by Govt. of Maharashtra)

## Aim:-

To design and develop a **Full Stack Finance Tracker Application** that enables users to manage their income, expenses, and savings efficiently, using modern web technologies like **React, Tailwind CSS, MongoDB, Node.js**, and integrated DevOps practices for **secure deployment and continuous integration**.

## 1. Introduction

Financial management is a critical aspect of personal and business life. Traditionally, people rely on manual record-keeping or spreadsheets, which are inefficient and prone to data loss or inconsistency. With the evolution of web technologies, there is a growing demand for **digital platforms** that provide automation, accuracy, and real-time insights into financial activities.

The project focuses on implementing **modern web technologies** to create a responsive user interface, a reliable backend API, and a scalable database. It integrates **authentication mechanisms, real-time data updates, and automated deployment pipelines**, following industry-standard software engineering practices.

This project integrates the complete full-stack development cycle — from **frontend UI to backend APIs, authentication, database management, and automated deployment pipelines** — following modern software development practices.

Key highlights include:

- **Responsive UI** built with **Tailwind CSS** and **React Hooks**.
- **State management** using **Context API / Redux**.
- **Secure backend APIs** with **JWT Authentication**.
- **Database management** with **MongoDB & Mongoose**.
- **Real-time updates** using **WebSockets**.
- **Deployment & CI/CD** using **Docker, GitHub Actions, and Render/Vercel**.

## 2. Literature Review

Several finance management tools exist today, such as **Mint**, **YNAB (You Need a Budget)**, and **PocketGuard**, which offer budgeting and expense-tracking features. These applications inspired the development of this project, but each has certain limitations — such as limited customization, paid features, or lack of open-source accessibility.

The **Finance Tracker** project aims to overcome these challenges by creating a **simplified, open-source alternative** that can be easily customized and deployed by users or developers. The project follows the **MERN stack architecture**, which is widely adopted in modern full-stack web development for its seamless JavaScript integration across the frontend, backend, and database layers.

In terms of academic grounding, several research studies and documentation (e.g., from MDN, MongoDB, React Docs, and Docker) were reviewed to ensure adherence to:

- RESTful API design principles.
- Secure authentication using **JWT** and **bcrypt**.
- Responsive UI design with **Tailwind CSS**.
- Continuous integration and deployment pipelines using **GitHub Actions**.

The combination of these modern frameworks and tools aligns with current software development trends, providing a strong foundation for building scalable, secure, and user-friendly applications.

## 3. Resources Required

### Hardware Requirements

- Computer/Laptop with minimum **8 GB RAM**
- Processor: Intel i5 or higher / AMD equivalent
- Stable internet connection
- Docker-compatible operating system (Windows, macOS, or Linux)

## Software Requirements

- **Frontend:** React.js (v18+), Tailwind CSS
- **Backend:** Node.js (v20+), Express.js
- **Database:** MongoDB (Atlas or local), Mongoose ORM
- **Authentication:** JSON Web Token (JWT), bcrypt
- **Testing Tools:** Postman for API validation
- **Real-time Communication:** Socket.io (WebSockets)
- **Version Control:** Git, GitHub
- **Containerization & Deployment:** Docker, Render, or Vercel
- **CI/CD Tools:** GitHub Actions

## Development Tools

- Visual Studio Code (IDE)
- Node Package Manager (npm)
- Git CLI
- Docker Desktop

## 4. Flow of Making Project

### Step 1: Frontend Development

- Initialized React project using `create-react-app`.
- Configured **Tailwind CSS** for responsive, modern styling.
- Created modular components such as Dashboard, Transaction Form, and Expense Chart.
- Integrated **React Hooks** like `useState`, `useEffect`, and `useContext` to manage component states and side effects.
- Ensured mobile responsiveness and dynamic updates without reloading.

### Step 2: Backend Development

- Developed RESTful APIs using **Express.js** and **Node.js**.
- Implemented routes for user registration, login, and CRUD operations on transactions.
- Added middleware for authentication, validation, and error handling.
- Integrated **JWT** for secure access control and **bcrypt** for password encryption.



# VIVEKANAND EDUCATION SOCIETY

---

## INSTITUTE OF TECHNOLOGY (AUTONOMOUS)

(Affiliated to University of Mumbai, Approved by AICTE & Recognized by Govt. of Maharashtra)

### Step 3: Database Integration

- Designed data models for **Users** and **Transactions** using **Mongoose**.
- Established a connection with MongoDB Atlas.
- Implemented query optimizations for fast data retrieval and aggregation.

### Step 4: Real-Time Communication

- Integrated **Socket.io (WebSockets)** to enable instant data updates when new transactions are added or deleted.

### Step 5: API Testing and Validation

- Tested every route using **Postman** for different request types (GET, POST, PUT, DELETE).
- Checked response accuracy, error handling, and authorization logic.

### Step 6: State Management

- Implemented global state management using **Context API** or **Redux** to synchronize frontend data with backend updates.

### Step 7: Containerization & Deployment

- Created **Dockerfiles** for frontend and backend.
- Configured `docker-compose` for multi-container setup.
- Set up **GitHub Actions** workflow for CI/CD — automating build, test, and deployment stages.
- Deployed frontend on **Vercel** and backend on **Render**, linking them to MongoDB Atlas.



# VIVEKANAND EDUCATION SOCIETY

## INSTITUTE OF TECHNOLOGY (AUTONOMOUS)

(Affiliated to University of Mumbai, Approved by AICTE & Recognized by Govt. of Maharashtra)

## 5. OUTPUT

### Finance Tracker

Sign in to your account

Email address

you@example.com

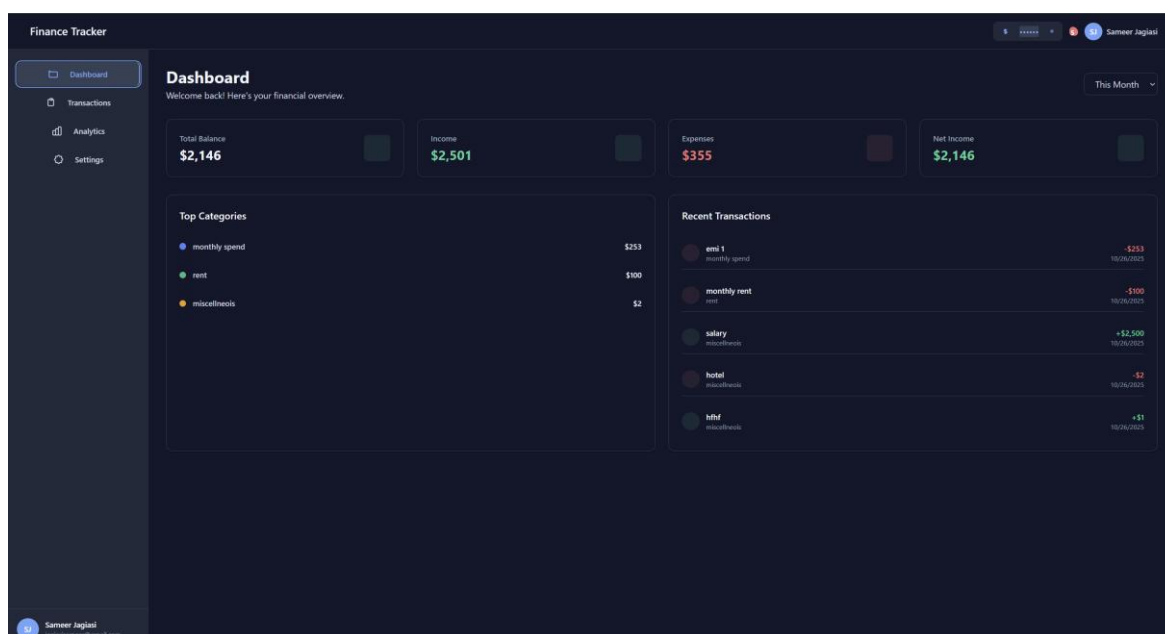
Password

\*\*\*\*\*

Sign in

Don't have an account?

Create a new account







# VIVEKANAND EDUCATION SOCIETY INSTITUTE OF TECHNOLOGY (AUTONOMOUS)

(Affiliated to University of Mumbai, Approved by AICTE & Recognized by Govt. of Maharashtra)

Finance Tracker

\$ \$2,146

SJ Sameer Jagiasi

Dashboard

Transactions

Analytics

Settings

Transactions

Manage your income and expenses

+ Add Transaction

Search

Type

Sort by

Search transactions...

All

Date

Descending

5 transactions

emi 1

monthly spend

-\$253

10/26/2025

monthly rent

rent

-\$100

10/26/2025

salary

miscellaneous

+\$2,500

10/26/2025

hotel

miscellaneous

-\$2

10/26/2025

hfhf

miscellaneous

+\$1

10/26/2025

Finance Tracker

\$ \$2,146

SJ Sameer Jagiasi

Dashboard

Transactions

Analytics

Settings

Add New Transaction

Type

Income

Category

e.g., Salary, Food, Transport

Amount

0.00

Description

Transaction description

Cancel

Add Transaction

+ Add Transaction

Sort by

Date

-\$253

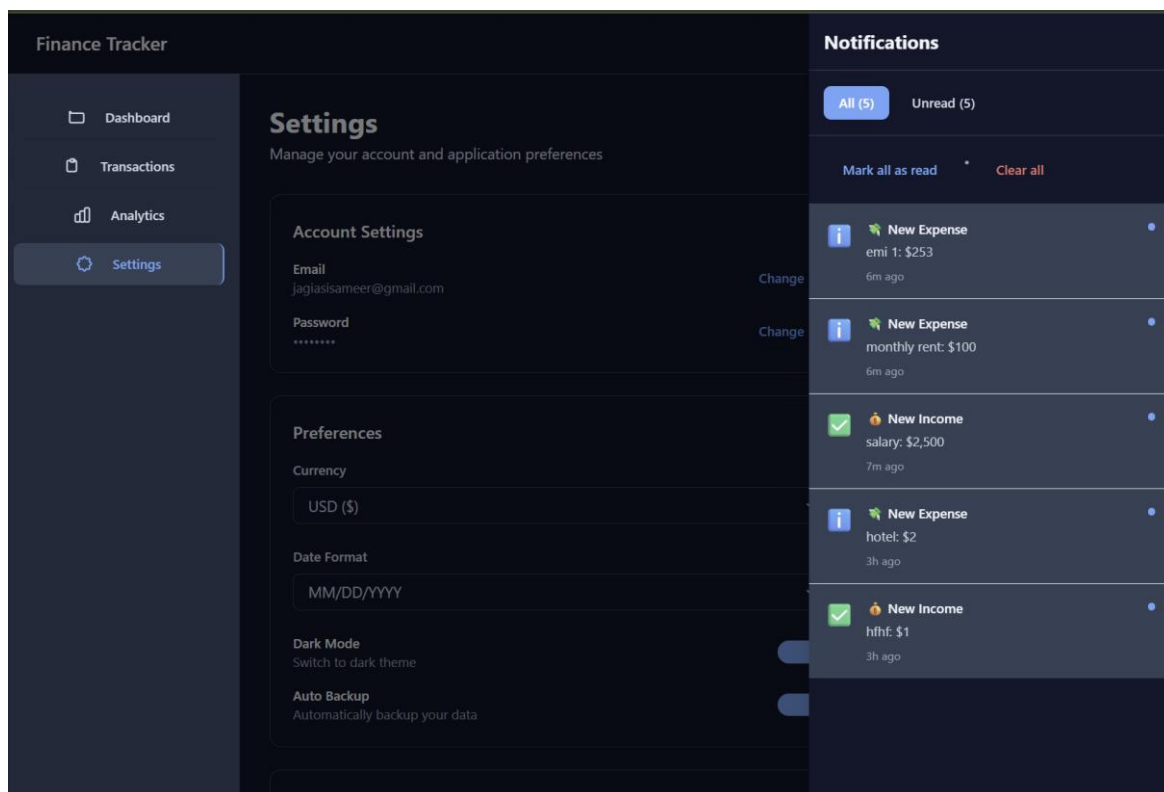
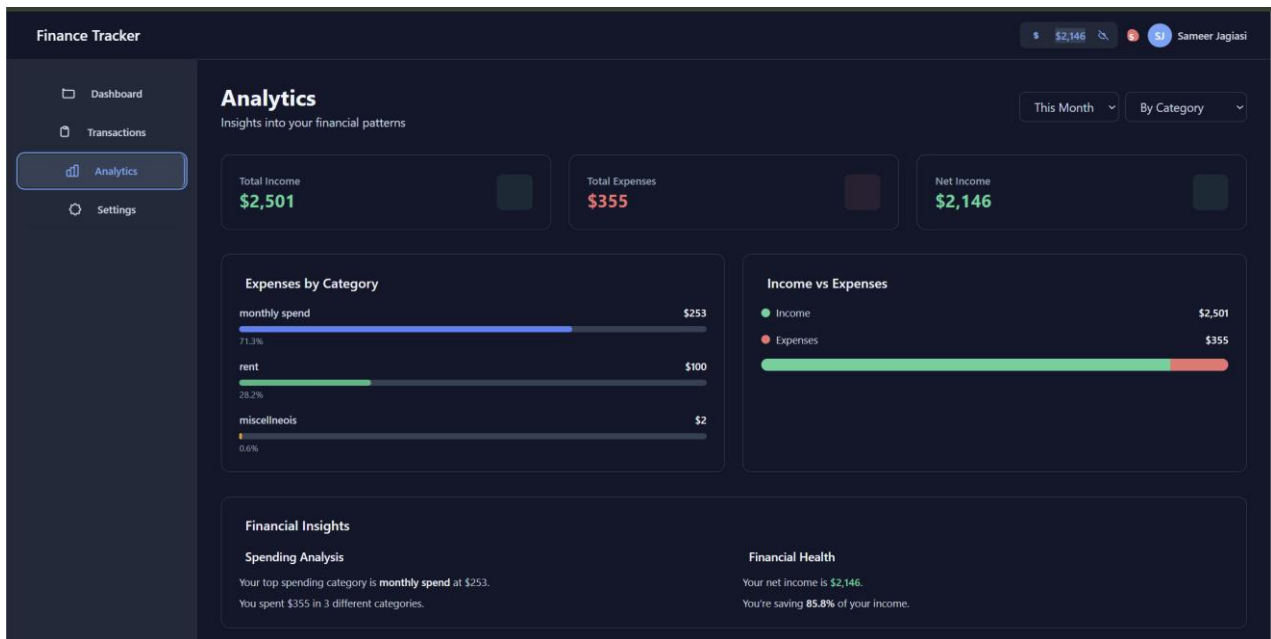
10/26/2025





# VIVEKANAND EDUCATION SOCIETY INSTITUTE OF TECHNOLOGY (AUTONOMOUS)

(Affiliated to University of Mumbai, Approved by AICTE & Recognized by Govt. of Maharashtra)

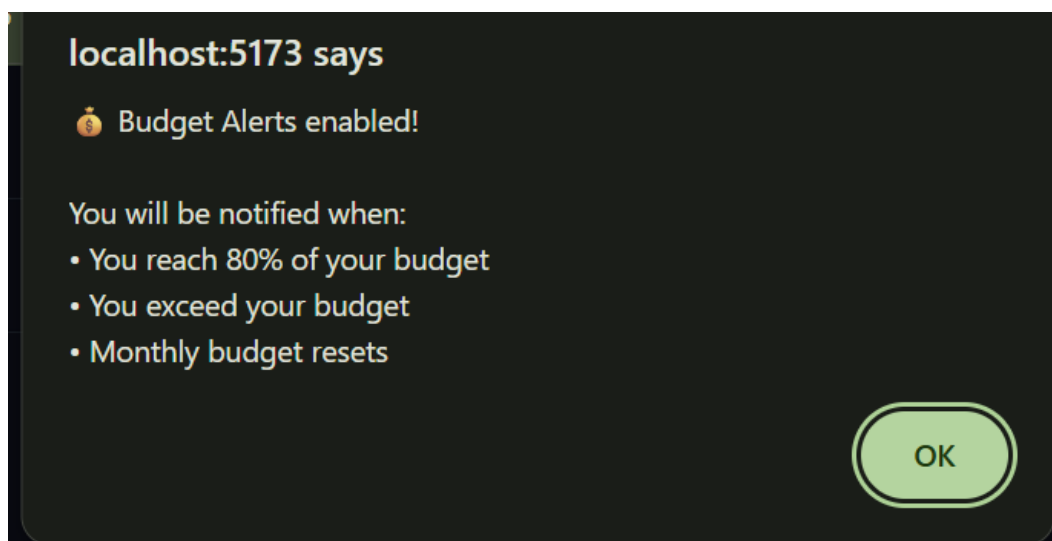
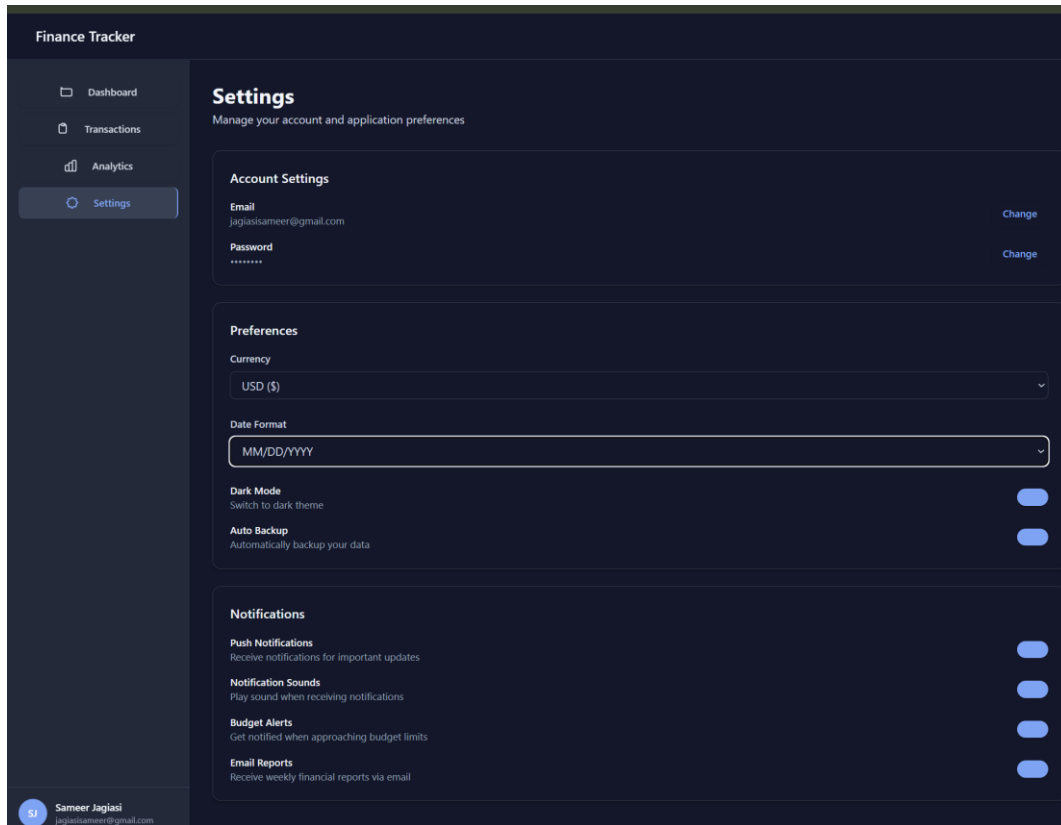




# VIVEKANAND EDUCATION SOCIETY

## INSTITUTE OF TECHNOLOGY (AUTONOMOUS)

(Affiliated to University of Mumbai, Approved by AICTE & Recognized by Govt. of Maharashtra)





# VIVEKANAND EDUCATION SOCIETY INSTITUTE OF TECHNOLOGY (AUTONOMOUS)

(Affiliated to University of Mumbai, Approved by AICTE & Recognized by Govt. of Maharashtra)

## 6. Challenges & Solutions

Challenge	Solution
CORS (Cross-Origin Resource Sharing) errors during frontend-backend integration	Configured CORS middleware in Express and proxy in React development server.
Handling user authentication securely	Implemented JWT tokens with bcrypt hashing and used HTTP-only cookies.
State synchronization issues	Applied Context API with reducer pattern for predictable state flow.
Docker networking between multiple containers	Used Docker Compose to define service dependencies and shared networks.
Managing CI/CD automation	Created GitHub Actions YAML pipeline with build and deployment triggers.
API performance optimization	Applied indexing in MongoDB and used async/await for non-blocking calls.

## 7. Future Enhancements

- **Data Visualization:** Integrate advanced charts and analytics (e.g., Recharts or Chart.js).
- **AI-based Expense Prediction:** Implement machine learning models for predictive budgeting.
- **Multi-user Roles:** Add admin and premium user features for business-level access.
- **Cloud Notifications:** Use Firebase Cloud Messaging or email alerts for reminders.
- **Mobile App Integration:** Convert the web app into a Progressive Web App (PWA) or develop a React Native version.
- **Currency Conversion and API Integration:** Support multiple currencies and global finance APIs.
- **Offline Mode:** Enable data caching for offline access using service workers.



# VIVEKANAND EDUCATION SOCIETY

---

## INSTITUTE OF TECHNOLOGY (AUTONOMOUS)

(Affiliated to University of Mumbai, Approved by AICTE & Recognized by Govt. of Maharashtra)

## 8. Conclusion

- The **Finance Tracker Full Stack Project** successfully demonstrates the complete lifecycle of a modern web application — from user interface design and backend API creation to real-time functionality and automated deployment.
- This project deepened understanding in **React, Node.js, MongoDB, DevOps, and CI/CD pipelines**, showcasing a true integration of full-stack and DevOps skills.
- It provides a reliable, scalable, and secure solution for financial tracking, with a clean and responsive user experience. The project also emphasizes best practices like code modularity, RESTful principles, version control, and container-based deployment.
- Overall, the Finance Tracker serves as a strong representation of **end-to-end software engineering principles**, reflecting both technical proficiency and real-world applicability.

## 9. References

- React Official Documentation. *React – A JavaScript library for building user interfaces.* <https://reactjs.org/docs/getting-started.html>
- Tailwind CSS Documentation. *Tailwind CSS – Utility-first CSS framework.* <https://tailwindcss.com/docs>
- Node.js Official Documentation. *Node.js® JavaScript runtime.* <https://nodejs.org/en/docs/>
- Express.js Documentation. *Web framework for Node.js.* <https://expressjs.com/>
- JWT Official Guide. *JSON Web Token (JWT) Authentication.* <https://jwt.io/introduction/>
- MongoDB Documentation. *MongoDB NoSQL database.* <https://www.mongodb.com/docs/>
- Mongoose Documentation. *MongoDB object modeling for Node.js.* <https://mongoosejs.com/docs/guide.html>
- Postman Learning Center. *API testing and validation tool.* <https://learning.postman.com/docs/getting-started/introduction/>
- Docker Documentation. *Containerization guide for applications.* <https://docs.docker.com/>
- GitHub Actions Documentation. *Automate CI/CD pipelines.* <https://docs.github.com/en/actions>



# **VIVEKANAND**

## **EDUCATION SOCIETY**

---

### **INSTITUTE OF TECHNOLOGY**

#### **(AUTONOMOUS)**