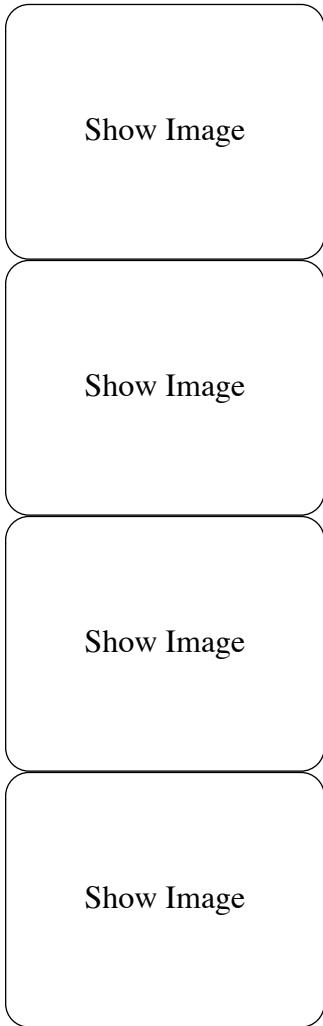




# iSteam Studio - Employee Hours Management System

A modern, full-stack employee time tracking and management system built with React, Flask, and PostgreSQL.



Show Image

Show Image

Show Image

Show Image

## ⭐ Features

### ⌚ Time Tracking

- **Clock In/Out** - Simple one-click time tracking
- **Real-time Timer** - Live display of worked hours
- **Break Management** - Track break time separately
- **Automatic Calculations** - Total hours computed automatically

### 👤 Manager Approval System

- **Email Notifications** - Managers notified on new entries
- **Approval Workflow** - Review and approve/deny time entries
- **Bulk Actions** - Process multiple entries efficiently
- **Status Tracking** - Track pending, approved, and denied entries

# Analytics Dashboard

- **Monthly Reports** - Total hours and employee statistics
- **Active Employees** - Track workforce engagement
- **Pending Queue** - Monitor approval backlog
- **Visual Analytics** - Easy-to-read metrics

## Security Features

- **JWT Authentication** - Secure token-based auth
- **Role-Based Access** - Employee, Manager, and Admin roles
- **Password Hashing** - Bcrypt encryption
- **MFA Ready** - Multi-factor authentication support

## Additional Features

- **Edit Requests** - Employees can request time corrections
- **Audit Trail** - Complete history of all actions
- **Timezone Support** - Handle multiple timezones
- **Email Integration** - Automated notifications

## Tech Stack

### Frontend

- **React 18.2** - Modern UI library
- **Tailwind CSS** - Utility-first styling
- **Lucide React** - Beautiful icons
- **Responsive Design** - Mobile-friendly interface

### Backend

- **Flask 3.0** - Python web framework
- **PostgreSQL** - Relational database
- **SQLAlchemy** - ORM for database operations
- **Flask-JWT-Extended** - JWT authentication
- **Flask-Bcrypt** - Password hashing

## Infrastructure

- **Gunicorn** - WSGI HTTP server
- **CORS** - Cross-origin resource sharing
- **SMTP** - Email notifications

## Prerequisites

- **Python 3.8+**
- **Node.js 16+** and npm
- **PostgreSQL 14+**
- **Git**

## Quick Start

### 1. Clone the Repository

```
bash
git clone https://github.com/YOUR_USERNAME/isteam-employee-hours.git
cd isteam-employee-hours
```

### 2. Setup Backend

```
bash
cd backend

# Create virtual environment
python -m venv venv
source venv/bin/activate # On Windows: venv\Scripts\activate

# Install dependencies
pip install -r requirements.txt

# Setup environment variables
cp .env.example .env
# Edit .env with your database credentials

# Initialize database
python app.py
# Visit http://localhost:5000/api/init-db in browser
```

### 3. Setup Frontend

```
bash
```

```
cd frontend
```

```
# Install dependencies
```

```
npm install
```

```
# Start development server
```

```
npm start
```

## 4. Access the Application

- **Frontend:** <http://localhost:3000>
- **Backend API:** <http://localhost:5000>

## 5. Test Login Credentials

- **Employee:** `(alex@isteam.com)` / `(password)`
- **Manager:** `(manager@isteam.com)` / `(password)`

## Project Structure

```
isteam-employee-hours/
├── backend/
│   ├── app.py          # Main Flask application
│   ├── requirements.txt # Python dependencies
│   ├── .env            # Environment variables
│   ├── Procfile        # Heroku deployment
│   └── render-build.sh # Render deployment
├── frontend/
│   ├── public/
│   │   └── index.html    # HTML template
│   └── src/
│       ├── App.js        # Main React component
│       ├── index.js      # React entry point
│       └── index.css     # Global styles
└── docs/
    ├── SETUP_GUIDE.md    # Detailed setup instructions
    ├── DEPLOYMENT_GUIDE.md # Deployment options
    └── FREE_HOSTING_GUIDE.md # Free hosting guide
└── README.md           # This file
```

## Free Hosting Options

### Render.com (Recommended - FREE)

bash

# See docs/FREE\_HOSTING\_GUIDE.md for detailed steps

1. Push code to GitHub
2. Sign up at render.com
3. Create PostgreSQL database
4. Deploy backend web [service](#)
5. Deploy frontend static site

### Railway.app (FREE \$5 credit/month)

bash

1. Sign up at [railway.app](#)
2. Connect GitHub repository
3. Add PostgreSQL database
4. Deploy backend and frontend

For detailed deployment instructions, see:

- **FREE hosting:** [docs/FREE\\_HOSTING\\_GUIDE.md](#)
- **All options:** [docs/DEPLOYMENT\\_GUIDE.md](#)

## Database Schema

### Users Table

- id, name, email, password\_hash, role, timezone, created\_at

### Time Entries Table

- id, user\_id, date, clock\_in, clock\_out, break\_time, total\_hours, status, approved\_by, created\_at

### Edit Requests Table

- id, time\_entry\_id, user\_id, changes, reason, status, reviewed\_by, created\_at

### Audit Logs Table

- id, user\_id, action, details, ip\_address, timestamp

## Configuration

### Environment Variables (.env)

```
bash

# Database
DATABASE_URL=postgresql://user:password@localhost/isteam_hours

# Security
JWT_SECRET_KEY=your-super-secret-key-change-in-production

# Email (Optional)
SMTP_SERVER=smtp.gmail.com
SMTP_PORT=587
SENDER_EMAIL=your-email@gmail.com
SENDER_PASSWORD=your-app-password

# Flask
FLASK_ENV=development
FLASK_DEBUG=True
```

## Testing

### Backend Tests

```
bash

cd backend
python -m pytest tests/
```

### Frontend Tests

```
bash

cd frontend
npm test
```

## API Documentation

### Authentication

- [POST /api/auth/register](#) - Register new user
- [POST /api/auth/login](#) - User login

### Time Entries

- `POST /api/time-entries/clock-in` - Clock in
- `POST /api/time-entries/clock-out` - Clock out
- `GET /api/time-entries` - Get time entries

## Approvals (Manager only)

- `GET /api/approvals/pending` - Get pending approvals
- `PUT /api/approvals/:id` - Approve/deny entry

## Analytics (Manager only)

- `GET /api/analytics` - Get analytics data
- `GET /api/audit-log` - Get audit trail

For complete API documentation, see [docs/API.md](#)

## 🤝 Contributing

1. Fork the repository
2. Create a feature branch (`(git checkout -b feature/AmazingFeature)`)
3. Commit your changes (`(git commit -m 'Add some AmazingFeature')`)
4. Push to the branch (`(git push origin feature/AmazingFeature)`)
5. Open a Pull Request

## License

This project is licensed under the MIT License - see the LICENSE file for details.

## Team

Built for **iSteam Studio** - Empowering the next generation through STEAM education.

- Website: <https://www.isteamstudio.com/>
- Email: [info@isteamstudio.com](mailto:info@isteamstudio.com)

## Support

For help and documentation:

- **Setup Issues:** See [docs/SETUP\\_GUIDE.md](#)
- **Deployment Help:** See [docs/DEPLOYMENT\\_GUIDE.md](#)
- **Free Hosting:** See [docs/FREE\\_HOSTING\\_GUIDE.md](#)

## Roadmap

- Mobile app (React Native)
- Payroll integration
- GPS-based clock-in
- Advanced reporting
- Slack integration
- Multi-language support

## Acknowledgments

- iSteam Studio for the opportunity
- React team for amazing framework
- Flask team for excellent Python web framework
- PostgreSQL for robust database

---

Made with ❤️ for iSteam Studio

 Star this repo if you find it helpful!