

KẾ HOẠCH THỰC HIỆN CHI TIẾT - SSB 1.0

✓ ĐÃ HOÀN THÀNH (Tuần 1-4)

- Task 1: Requirements Analysis
- Task 2: System Modeling
- Task 3: Architecture Design
- Task 4: Frontend Development (Next.js)

⌚ ĐANG THỰC HIỆN - TUẦN 5: BACKEND FOUNDATION

Thứ 2: Setup Backend Project

- Tạo thư mục `backend/`
- Init npm: `npm init -y`
- Cài packages cần thiết:

bash

```
npm install express mysql2 dotenv cors
npm install jsonwebtoken bcryptjs
npm install socket.io
npm install nodemon --save-dev
```

- Tạo file `server.js` cơ bản
- Setup `.env` với thông tin database

Thứ 3: Database Setup

- Tạo database MySQL: `CREATE DATABASE ssb_db;`
- Chạy script tạo bảng (sử dụng schema đã thiết kế)
- Tạo file `config/database.js` để kết nối MySQL
- Test connection thành công

Thứ 4: Authentication System

- Tạo models: `User.js`
- Tạo `authController.js`:
 - Register function (hash password với bcrypt)
 - Login function (verify & tạo JWT token)
- Tạo `routes/auth.js`
- Tạo middleware `auth.js` để verify token

- Test với Postman: register, login

Thứ 5: User & Driver Management

- Tạo models: `Driver.js`
- Tạo `userController.js`: CRUD users
- Tạo `driverController.js`: CRUD drivers
- Tạo routes tương ứng
- Test API với Postman

Thứ 6: Bus & Route Management

- Tạo models: `Bus.js`, `Route.js`
- Tạo `busController.js`: CRUD buses
- Tạo `routeController.js`: CRUD routes + stops
- Tạo routes tương ứng
- Test API

Thứ 7: Student & Schedule Management

- Tạo models: `Student.js`, `Schedule.js`
- Tạo `studentController.js`: CRUD students
- Tạo `scheduleController.js`: CRUD schedules
- Logic phân công học sinh vào lịch
- Test API

Chủ nhật: Review & Integration Test

- Test toàn bộ API đã tạo
- Fix bugs nếu có
- Chuẩn bị data mẫu (hardcode) để test

Output Tuần 5: Backend API cơ bản hoạt động đầy đủ

TUẦN 6: REAL-TIME TRACKING & INTEGRATION

Thứ 2: Socket.IO Setup

- Cài socket.io cho backend và frontend
- Setup Socket.IO server trong `server.js`
- Tạo `socket/trackingSocket.js`
- Test connection giữa client-server

Thứ 3: GPS Tracking API

- Tạo model `Tracking.js`
- Tao `trackingController.js`:
 - POST `/api/tracking/location` (lưu GPS)
 - GET `/api/tracking/schedule/:id` (lấy vị trí)
- Socket emit location update
- Test API

Thứ 4: Frontend Integration - Part 1

- Tạo axios config trong Next.js
- Tích hợp API authentication
- Tạo AuthContext/Provider
- Test login flow

Thứ 5: Frontend Integration - Part 2

- Connect Admin Dashboard với API
- Hiển thị data từ API: buses, drivers, students
- CRUD operations cho Admin

Thứ 6: Frontend Integration - Part 3

- Connect Driver interface với API
- Hiển thị lịch làm việc
- Connect Parent interface
- Hiển thị thông tin xe của con

Thứ 7: Real-time Map Integration

- Tích hợp Socket.IO vào Next.js
- Update map real-time khi nhận GPS data
- Test với mock GPS data
- Geofencing: thông báo khi xe gần

Chủ nhật: Testing Real-time Features

- Simulate: Driver gửi location
- Parent nhận update real-time
- Fix latency issues
- Optimize performance

Output Tuần 6: Real-time tracking hoạt động

TUẦN 7: ADVANCED FEATURES

Thứ 2: Notification System

- Tạo model `Notification.js`
- Tạo `notificationController.js`
- Socket.IO cho real-time notifications
- Test gửi thông báo

Thứ 3: Message System

- Tạo model `Message.js`
- Tạo `messageController.js`
- Giao diện chat đơn giản
- Test messaging

Thứ 4: Incident Reporting

- Tạo model `Incident.js`
- Tạo `incidentController.js`
- Driver có thể báo sự cố
- Admin nhận alert

Thứ 5: Pickup/Dropoff Tracking

- API đánh dấu đón/trả học sinh
- Update status trong database
- Parent nhận notification
- Test workflow

Thứ 6: Dashboard Statistics

- Tạo `dashboardController.js`
- API thống kê cho Admin
- Charts cho Dashboard
- Test data visualization

Thứ 7: Performance Optimization

- Database indexing
- API response caching
- Optimize Socket.IO
- Lazy loading frontend

Chủ nhật: Security & Bug Fixes

- Input validation tất cả API

- XSS protection
- SQL injection prevention
- Fix bugs

Output Tuần 7: Ứng dụng hoàn thiện

TUẦN 8: TESTING & DEPLOYMENT

Thứ 2: Testing

- Test toàn bộ user flows
- Test với 50-100 concurrent users
- Performance testing
- Load testing (optional)

Thứ 3: Documentation

- API Documentation (Postman/Swagger)
- User Manual (PDF)
- Code comments
- README.md

Thứ 4: Deployment Preparation

- Environment config (dev/prod)
- Database migration scripts
- Build Next.js: `npm run build`
- Test production build

Thứ 5: Deployment

- Deploy backend (Heroku/Railway/VPS)
- Deploy frontend (Vercel/Netlify)
- Configure CORS
- Test production

Thứ 6: Final Testing

- UAT (User Acceptance Testing)
- Fix critical bugs
- Performance check

Thứ 7: Presentation Prep

- Tạo slides PowerPoint

- Demo script
- Video demo (optional)

Chủ nhật: Final Presentation

- Trình bày dự án
- Live demo
- Q&A

Output Tuần 8: Dự án hoàn thiện & presentation

CÔNG NGHỆ SỬ DỤNG

Backend:

- **ExpressJS** - REST API server
- **MySQL** - Database
- **Socket.IO** - Real-time communication
- **JWT** - Authentication
- **Bcrypt** - Password hashing

Frontend:

- **Next.js** - React framework
- **React Leaflet** - Maps
- **Axios** - HTTP client
- **Socket.IO Client** - Real-time
- **Material-UI/Tailwind** - Styling

Tools:

- **Postman** - API testing
 - **MySQL Workbench** - Database management
 - **Git/GitHub** - Version control
 - **VS Code** - Code editor
-

TIPS CHO SINH VIÊN

1. **Làm từng bước nhỏ** - Không vội hoàn thành tất cả

2. **Test ngay sau khi code** - Đừng đợi đến cuối
 3. **Commit Git thường xuyên** - Mỗi tính năng 1 commit
 4. **Hardcode data để test nhanh** - Không cần data thật ban đầu
 5. **Tập trung vào core features** - Skip features phức tạp nếu thiếu thời gian
 6. **Hỏi khi gặp khó khăn** - Đừng stuck quá lâu
 7. **Document code** - Viết comment để nhớ logic
-

PRIORITY FEATURES (Nếu thiếu thời gian)

MUST HAVE (Bắt buộc):

- Authentication
- CRUD: Buses, Drivers, Students, Routes
- Basic Schedule Management
- Simple GPS tracking (có thể fake)

NICE TO HAVE (Nên có):

- Real-time tracking với Socket.IO
- Notifications
- Dashboard statistics

OPTIONAL (Tùy chọn):

- Message system
- Incident reporting
- Advanced analytics
- Mobile app