

**Gamified VR-Based Therapy System for Upper Limb
Rehabilitation in Children with Hemiplegia**

25-26J-472

Deployment Report

B.Sc. (Hons) Degree in Information Technology specialized in
Software Engineering

Department of Software Engineering
Sri Lanka Institute of Information Technology
Sri Lanka

1. INTRODUCTION

The Gamified VR-Based Therapy System is an innovative rehabilitation solution designed for children with hemiplegia. The system integrates wearable sensors, machine learning, and virtual reality to provide engaging upper limb therapy. This report documents the deployment architecture, infrastructure, and implementation of the supporting software ecosystem, including mobile, web, and desktop applications for real-time patient progress tracking.

1.1 Deployment Scope

The deployment encompasses:

- Backend services and API infrastructure.
- Web application for doctors/therapists.
- Mobile application for patients/parents.
- Desktop application (Electron wrapper).
- Database management system.
- VPS hosting configuration.
- Security and access control implementation

2. TECHNOLOGY STACK DEPLOYMENT

Component	Technology	Deployment Method
Backend API	NestJS	PM2 Process Manager
Web Frontend	Next.js	Static Build + Nginx
Mobile App	React Native	Expo + App Stores
Desktop App	Electron.js	Packaged Installer
Database	PostgreSQL	Native Installation
Reverse Proxy	Nginx	System Service
SSL/TLS	Let's Encrypt	Certbot
Version Control	Git + GitHub	Repository

3. SERVER INFRASTRUCTURE

3.1 Hostinger VPS Configuration.

Server Specifications:

- **Provider:** Hostinger VPS
- **Operating System:** Ubuntu 22.04 LTS
- **CPU:** 8 vCPU Cores
- **RAM:** 8 GB
- **Storage:** 50 GB NVMe SSD
- **Bandwidth:** 4 TB/month

3.2 VPS Initial Setup (Hostinger)

```
# SSH into your Hostinger VPS
ssh root@your_vps_ip

# Update system
sudo apt update && sudo apt upgrade -y

# Install Node.js 18.x (NestJS requires 18+)
curl -fsSL https://deb.nodesource.com/setup_18.x | sudo -E bash -
sudo apt install -y nodejs

# Install build tools
sudo apt install -y build-essential git nginx

# Install PM2 globally
npm install -g pm2

# Verify installations
node --version
npm --version
pm2 --version
```

3.3 Database Deployment

PostgreSQL Setup

```
# Install PostgreSQL
sudo apt install -y postgresql postgresql-contrib

# Start PostgreSQL
sudo systemctl start postgresql
sudo systemctl enable postgresql

# Create database and user
sudo -u postgres psql
```

```
CREATE DATABASE rehab_db;
CREATE USER rehab_user WITH PASSWORD 'your_secure_password';
GRANT ALL PRIVILEGES ON DATABASE rehab_db TO rehab_user;
\q
```

3.4 Backend Deployment (Nest.Js)

```

# Clone backend repository
git clone https://github.com/IT22102546/Armigo-Research.git backend
cd backend

# Install dependencies
npm install

# Create environment file
cat > .env << EOF
NODE_ENV=production
PORT=3008
DATABASE_URL=postgresql://rehab.user:it22557292@localhost:5432/rehab_db
JWT_SECRET=$(openssl rand -hex 32)
CORS_ORIGIN=https://rehab.armigo.com,https://api.armigo.com
EOF

# Build the project
npm run build

# Start with PM2
pm2 start dist/main.js --name rehab-backend
pm2 startup
pm2 save

```

3.5 Web App Deployment(NexT.Js)

```

cd /var/www/rehab-system
git clone https://github.com/IT22102546/Armigo-Research.git webapp
cd webapp

# Install dependencies
npm install

# Create .env file
cat > .env.local << EOF
NEXT_PUBLIC_API_URL=https://api.armigo.com
NEXT_PUBLIC_WS_URL=wss://api.rehab.armigo.com
EOF

# Build the application
npm run build

# Export static files (if using static export)
npm run export
# The static files will be in 'out' directory.

```

3.6 Mobile App Deployment (React Native)

```
# Install Expo CLI
npm install -g eas-cli
eas build:configure

# Build for Android
eas build -p android --profile production

# Build for iOS (requires Mac)
eas build -p ios --profile production
```

3.7 Desktop App Build (Electron Js)

```
# Build Electron app
npm install

# Update API endpoint in .env
echo "API_URL=https://api.rehab.yourdomain.com" > .env

# Build for all platforms
npm run electron:build

# Build for specific platforms
npm run electron:build:win # Windows
npm run electron:build:mac # macOS
npm run electron:build:linux # Linux
```

4. NGINX CONFIGURATION

```
# Remove default nginx site
sudo rm /etc/nginx/sites-enabled/default

# Create backend API configuration
sudo nano /etc/nginx/sites-available/api.rehab.armago.com
```

```

server {
    listen 80;
    server_name api.rehab.arnigo.com;

    location / {
        proxy_pass http://localhost:3000;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_cache_bypass $http_upgrade;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;

        # Increased timeout for ML model processing
        proxy_read_timeout 300s;
        proxy_connect_timeout 300s;
    }
}

```

```

# Create web app configuration
sudo nano /etc/nginx/sites-available/rehab.arnigo.com

```

```

server {
    listen 80;
    server_name rehab.arnigo.com;
    root /var/www/rehab-system/webapp/out;
    index index.html;

    location / {
        try_files $uri $uri.html $uri/ -404;
    }

    # Cache static assets
    location /next/static {
        alias /var/www/rehab-system/webapp/next/static;
        expires 365d;
        add_header Cache-Control "public, immutable";
    }
}

```

```

# Enable sites
sudo ln -s /etc/nginx/sites-available/api.rehab.yourdomain.com /etc/nginx/sites-enabled/
sudo ln -s /etc/nginx/sites-available/rehab.yourdomain.com /etc/nginx/sites-enabled/

# Test nginx configuration
sudo nginx -t

# Restart nginx
sudo systemctl restart nginx

```

5. SSL CERTIFICATES

```

# Install certbot
sudo apt install -y certbot python3-certbot-nginx

# Obtain SSL certificates
sudo certbot --nginx -d rehab.armigo.com -d api.rehab.armigo.com

# Test auto-renewal
sudo certbot renew --dry-run

```

6. FIREWALL CONFIGURATION

```

# Configure UFW
sudo ufw allow OpenSSH
sudo ufw allow 'nginx full'
sudo ufw allow 3000/tcp # Backend API
sudo ufw allow 4000/tcp # Optional: if using additional services

# Enable firewall
sudo ufw --force enable

# Check status
sudo ufw status verbose

```


7. CONCLUSION

The deployment of the Gamified VR-Based Therapy System has been successfully completed on Hostinger VPS infrastructure. The system architecture provides:

1. Scalability: The microservices architecture allows independent scaling of components
2. Security: Multi-layered security with JWT authentication, rate limiting, and encrypted data storage
3. Reliability: PM2 process management ensures high availability with automatic recovery
4. Accessibility: Cross-platform support through web, mobile, and desktop applications
5. Monitoring: Comprehensive logging and monitoring for proactive issue detection

The deployed system meets all requirements specified in the research project and is ready for clinical trials and user acceptance testing.