🚀 EVLens Backend API Documentation

🛠️ Tech Stack

- Backend Framework: Node.js with Express
- Database: MongoDB with Mongoose
- Authentication: JWT + bcryptjs
- Templating (if needed): Handlebars (hbs)
- Deployment Ready: CORS for frontend integration with Verce

🔐 Environment Variables (Required in .env)

- PORT=your_desired_port
- MONGOURL=mongodb_connection_string
- KEY=jwt_secret_key

📂 Directory Structure

```
project/
│
├── public/          // Static assets
├── .env             // Environment variables
├── server.js
```

🔄 Middleware

🔑 verifyToken(req, res, next)

- Extracts JWT from Authorization header.
- Verifies token using process.env.KEY.
- Adds req.user with decoded data (username).

👤 User Authentication

📝 POST /api/signup

Creates a new user account.

Request Body:

JSON:

```
{
  "name": "username",
  "password": "plaintextpassword"
}
```

Response:

- 201 Created: { message: "Successfully created account." }
- 400 Bad Request: { error: "Username not available" }
- 500 Server Error

🔑 POST /api/login

Authenticates user and returns a JWT.

Request Body:

json:

```
{
 "name": "username",
"password": "plaintextpassword"
}
```

Response:

json:

```
{
 "status": "Success",
"token": "JWT_TOKEN"
}
```

- 401 Unauthorized: Incorrect credentials
- 500 Server Error

⚡ Station Management (Protected Routes)

All following routes require Authorization: Bearer <token> in headers.

➕ POST /api/AddStation

Adds a new station.

Request Body:

json:

```
{
 "Name": "StationName",
 "Latitude": 19.123,
 "Longitude": 72.456,
 "Status": "Active",
 "PowerOutput": 120,
 "ConnectorType": "Type2"
}
```

Response:

- 200 OK: { Data: "Successfully Added" }
- 400 Bad Request or 409 Conflict: Validation errors
- 500 Server Error

❌ POST /api/DeleteStation

Deletes a station by name (only if added by the user).

Request Body:

json:

```
{
 "Name": "StationName"
}
```

Response:

- 200 OK: { Data: "Successfully Deleted the Station" }
- 404 Not Found
- 500 Server Error

♻️ POST /api/UpdateStation

Updates a station's details (must belong to user).

Request Body:

json:

```
{
 "Name": "StationName",
 "Latitude": 19.123,
 "Longitude": 72.456,
 "Status": "Inactive",
 "PowerOutput": 100,
 "ConnectorType": "CCS"
}
```

Response:

- 200 OK: { Data: "Successfully Updated" }
- 404 Not Found
- 400 Bad Request
- 500 Server Error

📥 GET /api/GetStations

Fetches all stations added by the authenticated user.

Response:

json:

```
[ {
 "Name": "Station 1",
 "Latitude": 19.1,
 "Longitude": 72.8, ...
 }]
```

📊 User Activity Log

📖 GET /api/GetActivity

Returns latest 10 activities performed by user.

Response:

json:

```
[ {
 "user": "john",
 "action": "Added",
 "stationName": "Station 1",
 "timestamp": "2025-06-06T12:00:00Z"
}, ... ]
```

🧱 Models

User:

```
{
 name: String,   // Unique
 password: String // Hashed
}
```

```
Station:
{
  Name: String,
  Latitude: Number,
  Longitude: Number,
  Status: String,
  PowerOutput: Number,
  ConnectorType: String,
  AddedBy: String
}

Activity:
{
  user: String,
  action: "Added" | "Updated" | "Deleted",
  stationName: String,
  timestamp: Date
}
```

🔐 Security Features

- JWT Auth with middleware validation
- Password Hashing using bcryptjs
- CORS Configuration for https://evlens.vercel.app
- Protected CRUD operations based on user identity
