# API Documentation

## Overview

This API is built using Express, TypeScript, Prisma, and includes authentication with Passport.js.

## Server Configuration

Port: Configured via environment variable, defaults to 3000

CORS: Configured for http://localhost:3000

Middleware:

- JSON parsing

- URL-encoded body parsing

- Session management

- Passport authentication

## Create a Problem

Endpoint: POST /problem/create

Request Body:

```
{
  "title": string,
  "description": string,
  "category": string,
  "budget": number,
  "status": string,
  "userID": number
}
```

Responses:

201: Problem created successfully

```
{
  "message": "Problem created successfully",
  "problem": {
    "problemID": number,
    "title": string,
    "description": string,
    "category": string,
    "budget": number,
    "status": string,
    "userID": number
  }
}
```

400: Missing required fields

500: Server error when creating problem

## Update a Problem

Endpoint: PUT /problem/update/:problemID

Request Parameters:

problemID: Numeric ID of the problem

Request Body:

```
{
  "title"?: string,
  "description"?: string,
  "category"?: string,
  "budget"?: number,
  "status"?: string
}
```

Responses:

200: Problem updated successfully

```
{
  "message": "Problem updated successfully",
  "problem": {
    "problemID": number,
    "title": string,
    "description": string,
    "category": string,
    "budget": number,
    "status": string
  }
}
```

500: Server error when updating problem

## Attach File to Problem

Endpoint: POST /problem/attach-file/:problemID

Request Parameters:

problemID: Numeric ID of the problem

Request Body:

Multipart form-data with file upload

Responses:

200: File attached successfully

```
{
  "message": "File attached successfully",
  "problem": {
    "attachments": string (file path)
  }
}
```

400: No file uploaded

500: Error attaching file

## Add Tags to Problem

Endpoint: POST /problem/add-tags/:problemID

Request Parameters:

problemID: Numeric ID of the problem

Request Body:

```
{
  "tags": string[]
}
```

Responses:

200: Tags added successfully

```
{
  "message": "Tags added successfully",
  "problem": {
    "category": string (comma-separated tags)
  }
}
```

400: Invalid tags format

500: Error adding tags

### Get Problem Details

Endpoint: GET /problem/:problemID

Request Parameters:

problemID: Numeric ID of the problem

Responses:

200: Problem details retrieved

```
{
  "problemID": number,
  "title": string,
  "description": string,
  "category": string,
  "budget": number,
  "status": string,
  "userID": number,
  "attachments"?: string
}
```

404: Problem not found

500: Error retrieving problem