Backend API Framework

Define the APIs you expect in your backend service (RESTful services). Each API should include the URI, HTTP method, request params/body, and response code/body.

AUTH ROUTES (/auth)	
Resource Operation **HTTP Method** **URI (HTTP)**	**Request Parameter or Data** **Response Code**
Register an account **POST** **/register**	**Request Parameter** { **“data”: {** **“firstName”**: *<string>*, **“lastName”**: *<string>*, **“email”**: *<string>*, **“password”**: *<string>*, **“serverPath”**: “/users”, **“picturePath”**: *<string>*, **“picture”**: *<image>*, **“location”**: *<string>*, **“occupation”**: *<string>*, **}}** **Response Code** **Success** **{ “status”**: 201, **“data”: { “message”**: “Successfully registered user!” **}}** **Failure** **{ “status”**: 409, **“data”: { “message”**: ”This email is already in use!” **}}** **{ “status”**: 500, **“data”: { “message”**: *<err.message>* **}}**

Login (authenticates user and gets a token)
 	
POST
 	
/login
	Request Parameter
	{
	“headers”: { "Content-Type": "application/json" **},**
	“data”: { “email”: *“example@gmail.com”* **“password”**: *“eXaMp!e11”* **}**
	}
	
	Response Code
	Success
	{ “status”: 200**,**
	“data”: {
	“token”: *<some token>*
	“user”: {

	“_id”: *<some id>*, **“firstName”:** *<string>*, **“lastName”:** *<string>*, **“email”:** *<string>*, **“password”:** *<string>,* **“picturePath”:** *<string>*, **“friends”:** *<array of user ids>*, **“location”:** *<string>*, **“occupation”:** *<string>*, **“viewedProfile”:** *<number>*, **“impressions”:** *<number>*, **“twitterHandle”:** *<string>*, **“linkedInHandle”:** *<string>*, **}}}** **Failure** **{ “status”: 400, “data”: { “message”:***“Invalid Credentials - Wrong Email or Password”* **}}** **{ “status”: 500, “data”: { “message”:** *<err.message>* **}}**

Backend API Framework

USERS ROUTES (/users)	
Resource Operation **HTTP Method** **URI (HTTP)**	**Request Parameter or Data** **Response Code**
Get all users **GET** **/**	**Response Code** Success **{ “status”: 200, “data”: { “allUsers”:** <array of user data objects> **}}** User Data Object Model: **{ _id, firstName, lastName, picturePath }** Failure **{ “status”: 404, “data”: { “message”:** *<err.message>* **}}**
Get User **GET** **/:id**	**Response Code** Success **{ “status”: 200,** **“data”: {** **“user”: {** **“_id”:** *<some id>*, **“firstName”:** *<string>*, **“lastName”:** *<string>*, **“email”:** *<string>*, **“password”:** *<string>*, **“picturePath”:** *<string>*, **“friends”:** *<array of user ids>*, **“location”:** *<string>*, **“occupation”:** *<string>*, **“viewedProfile”:** *<number>*, **“impressions”:** *<number>*, **“twitterHandle”:** *<string>*, **“linkedInHandle”:** *<string>*, **}}}** Failure **{ “status”: 404, “data”: { “message”:** *<err.message>* **}}**
Get User Friends **GET** **/:id/friends**	**Response Code** Success **{ “status”: 200, “data”: { “friends”:** <array of user’s friends> **}}** User Friend Data Object Model:

	{ _id, firstName, lastName, occupation, location, picturePath } **Failure** **{ “status”: 404, “data”: { “message”:** *<err.message>* **}}**
Add/Remove Friend **PATCH** **/:id/friends**	**Request Parameter** **{ “headers”: { “Authorization”:** \`Bearer ${token}\`, **"Content-Type":** "application/json" **} }** **Response Code** **Success** **{ “status”: 200, “data”: { “loggedInUserFriends”:** <array of friends>**, “otherUsersFriends”:** *<array of friends>* **}}** Friend Data Object Format **{ _id, firstName, lastName, occupation, location, picturePath }** **Failure** **{ “status”: 404, “data”: { “message”:** *<err.message>* **}}** **{ “status”: 405, “data”: { “message”:** “You cannot add yourself as a friend.” **}}**
Update User Information **PATCH** **/:id**	**Request Parameter** **{** **“headers”: { “Authorization”:** \`Bearer ${token}\` **}** **“data”: {** **“_id”:** *<some id>*, **“firstName”:** *<string>*, **“lastName”:** *<string>*, **“serverPath”:** “/users”, **“picturePath”:** *<string>*, **“picture”:** *<image>*, **“location”:** *<string>*, **“occupation”:** *<string>*, **“twitterHandle”:** *<string>*, **“linkedInHandle”:** *<string>*, **}}** **Response Code** **Success** **{ “status”: 200,** **“data”: {** **“updatedUser”: {** **“_id”:** *<some id>*, **“firstName”:** *<string>*, **“lastName”:** *<string>*, **“email”:** *<string>*, **“password”:** *<string>*, **“picturePath”:** *<string>*, **“friends”:** *<array of user ids>*, **“location”:** *<string>*, **“occupation”:** *<string>*, **“viewedProfile”:** *<number>*, **“impressions”:** *<number>*,

	“twitterHandle”: *<string>*, **“linkedInHandle”:** *<string>*,
	}}}
	
	Failure
	{ “status”: **404**, **“data”: { “message”:** *<err.message>* **}}**

Backend API Framework

POSTS ROUTES (/posts)	
Resource Operation **HTTP Method** **URI (HTTP)**	**Request Parameter or Data** **Response Code**
Create a post **POST** /	**Request Parameter** **{ “data”: {** **“userId”:** *<some user ID>*, **“description”:** *<string>*, **“serverPath”:** “/posts”, **“picturePath”:** *<string>*, **“picture”:** *<image>*, **“attachment”:** *<file>*, **“occupation”:** *<string>*, **}}** **Response Code** **Success** **{ “status”: 201,** **“data”: {** **“post”: {** **“_id”:** *<post ID>*, **“userId”:** *<user ID>*, **“description”:** *<string>*, **picturePath”:** *<string>*, **“likes”:** *<user ids who liked the post>*, **“comments”:** *<array of comment objects>* **}}}}** Comment Data Object Format **{ _id, userId, comment }** **Failure** **{ “status”: 409, “data”: { “message”:** *<err.message>* **}}**
Get all posts **GET** /	**Response Code** **Success** **{ “status”: 200, “data”: { “posts”:** <array of post data objects> **}}** Post Data Object Model: **{ _id, userId, description, picturePath, likes, comments }** **Failure**

Backend API Framework

	{ “status”: 404**, “data”: { “message”:** *<err.message>* **}}**
Get posts by a single user **GET** **/:userId/posts**	**Response Code** **Success** **{ “status”:** 200**, “data”: { “posts”:** <array of user’s post data objects> **}}** Post Data Object Model: **{ _id, userId, description, picturePath, likes, comments }** **Failure** **{ “status”:** 404**, “data”: { “message”:** *<err.message>* **}}**

Like/Dislike a post
 	
PATCH
 	
/:id/like
	Request Parameter
	{
	“headers”: { “Authorization”: \`Bearer ${token}\`, **"Content-Type":** "application/json" **}**
	“data”: { “userId”: *<user ID>* **}**
	}
	
	Response Code
	Success
	{ “status”: 200**,**
	“data”: {
	“post”: {
	“_id: *<post ID>*, **“userId”:** *<user ID>*, **“description”:** *<string>*, **picturePath”:** *<string>*,
	“likes”: *<user ids who liked the post>*, **“comments”:** *<array of comment objects>*
	}}}
	
	Failure
	{ “status”: 404**, “data”: { “message”:** *<err.message>* **}}**

Comment on a post
 	
PATCH
 	
/:id/comment
	Request Parameter
	{
	“headers”: { “Authorization”: \`Bearer ${token}\`, **"Content-Type":** "application/json" **}**
	“data”: { “userId”: *<user ID>*, **“comment”:** *<user comment>* **}**
	}

	Response Code
	Success
	{ “status”: 200,
	“data”: {
	“post”: {
	“_id: *<post ID>*, “userId”: *<user ID>*, “description”: *<string>*, picturePath”: *<string>*,
	“likes”: *<user ids who liked the post>*, “comments”: *<array of comment objects>*
	}}}
	
	Failure
	{ “status”: 404, “data”: { “message”: *<err.message>* }}

Delete a post
 	
DELETE
 	
/:id/delete
	Request Parameter
	{ “headers”: { “Authorization”: `Bearer ${token}`, "Content-Type": "application/json" } }
	
	Response Code
	Success
	{ “status”: 202,
	“data”: {
	“deletedPost”: {
	“_id: *<post ID>*, “userId”: *<user ID>*, “description”: *<string>*, picturePath”: *<string>*,
	“likes”: *<user ids who liked the post>*, “comments”: *<array of comment objects>*
	}}}
	
	Failure
	{ “status”: 404, “data”: { “message”: *<err.message>* }}

Backend API Framework

ADVERTISEMENTS ROUTES (/advertisements)

Resource Operation **HTTP Method** **URI (HTTP)**	**Request Parameter or Data** **Response Code**
Create an advert **POSTS** /	**Request Parameter** **{ “data”: {** **“name”:** *<string>*, **“website”:** *<string>*, **“description”:** *<string>*, **“serverPath”:** “/advertisements”, **“picturePath”:** *<string>*, **“picture”:** *<image>* **}}** **Response Code** **Success** **{ “status”:** 201 **}** **Failure** **{ “status”:** 409, **“data”: { “message”:** *<err.message>* **}}**
Get all advertisements **GET** /	**Request Parameter** **NONE** **Response Code** **Success** **{ “status”:** 200, **“data”: { “ads”:** <array of advertisement data objects> **}}** Advert Data Object Model: **{ name, website, picturePath, description }** **Failure** **{ “status”:** 404, **“data”: { “message”:** *<err.message>* **}}**