Animimic Web project – Release 1

## Readme:

We aim to create a website that integrates an animal behaviour simulation, allowing users to explore how animals can act like others in an interactive way. Additionally, the platform will have feature blogs and a discussion forum for knowledge sharing, and discussions around animal behaviour and related topics.

### Features:

1. Blog for Pet owners to communicate about their pets, pet foods, pet care, etc.

Users’ profiles

Parents and Kids: Assist children with using the simulation and reading educational content regarding the animals.

Animal Enthusiasts: Read blogs, participate in forums, and explore the simulation. They can also contribute by uploading any blogs or articles.

Pet Owners: They can share their personal experiences, stories, and insights about their pets' behaviours through blog posts or forum discussions.

Casual Visitors: Browse blogs and try out the simulation casually.

1. Animation on one animal behaving like other animal(s).

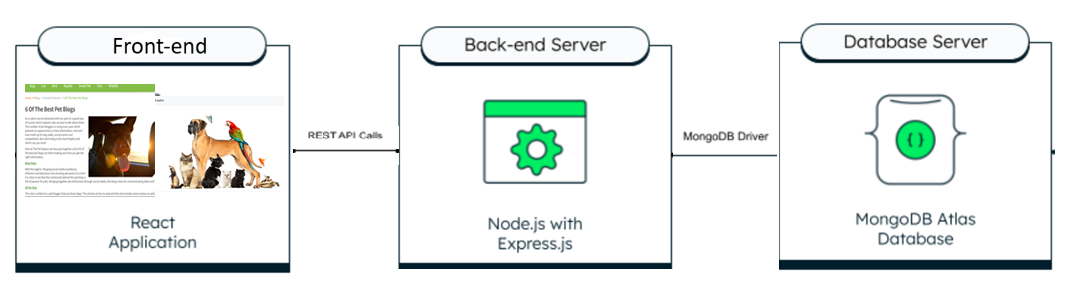
### Technology stack:

Frontend: Use of frameworks like React.js for a responsive design.

JavaScript libraries like Three.js for simulation of animal behaviour

Backend: Use of servers like Node.js and databases like MongoDB for user data and content management.

Implement 3D animation with Three.js



### Link to GitHub Repository:

Source codes of Front end for animation is included in the ‘webProject’ folder (<https://github.com/Srikrishna-madhusudhanan/Web-Development-Project/tree/main/webProject>)

Data base model for pet blog user registration is included in the 'Database' folder for release 1.

(<https://github.com/Srikrishna-madhusudhanan/Web-Development-Project/tree/main/Database>)

Web project – all documents reference:

<https://github.com/Srikrishna-madhusudhanan/Web-Development-Project/tree/main>

### Use case Diagrams:

<https://github.com/Srikrishna-madhusudhanan/Web-Development-Project/tree/main/SE/Requirements>

### Database Details

**DB & Interface:** Mongodb, Node JS/Express JS

**User data model for register, login, reset password and update profile details:**

const userSchema = new mongoose.Schema({

  name: String,

  mobile: String,

  username: {

    type: String,

    //required: true,

    unique: true

  },

  email: {

    type: String,

    //required: true,

    unique: true

  },

  password: String,

  secquestion: String,

  secanswer: String,

  photolink: String

});

#### 4.1 API Details:

##### 4.1.1 For new user registration:

POST <https://animimic.onrender.com/api/register>

##### 4.1.2 For user login:

POST <https://animimic.onrender.com/api/login>

Login will be based on username and password. On successful login, a token will be returned to the front-end in API response. The token has to be used in all future API calls except login and register APIs.

##### 4.1.3 For user details update

PUT <https://animimic.onrender.com/api/update>

Email is mandatory to identify the user in database table. Name, Mobile, Security Question(‘secquestion’) and Security Answer(‘secanswer’) fields can be updated through this update api call.

##### 4.1.4 For user password reset

POST <https://animimic.onrender.com/api/reset>

The ‘secanswer’ is mandatory field to validate before updating the new password for the user.

### Individual Contributions:

Dhanya: 3D animation

Pranathi: Pet Blog front-end

Sowmya Sri: Pet Blog front-end

Indusahithi: Pet Blog front-end

Niharika: Pet Blog document collection handling and photo handling for user profile

Srikrishna: REST API creation for pet Blog’s login, registration, password reset and user details update in MongoDB.