

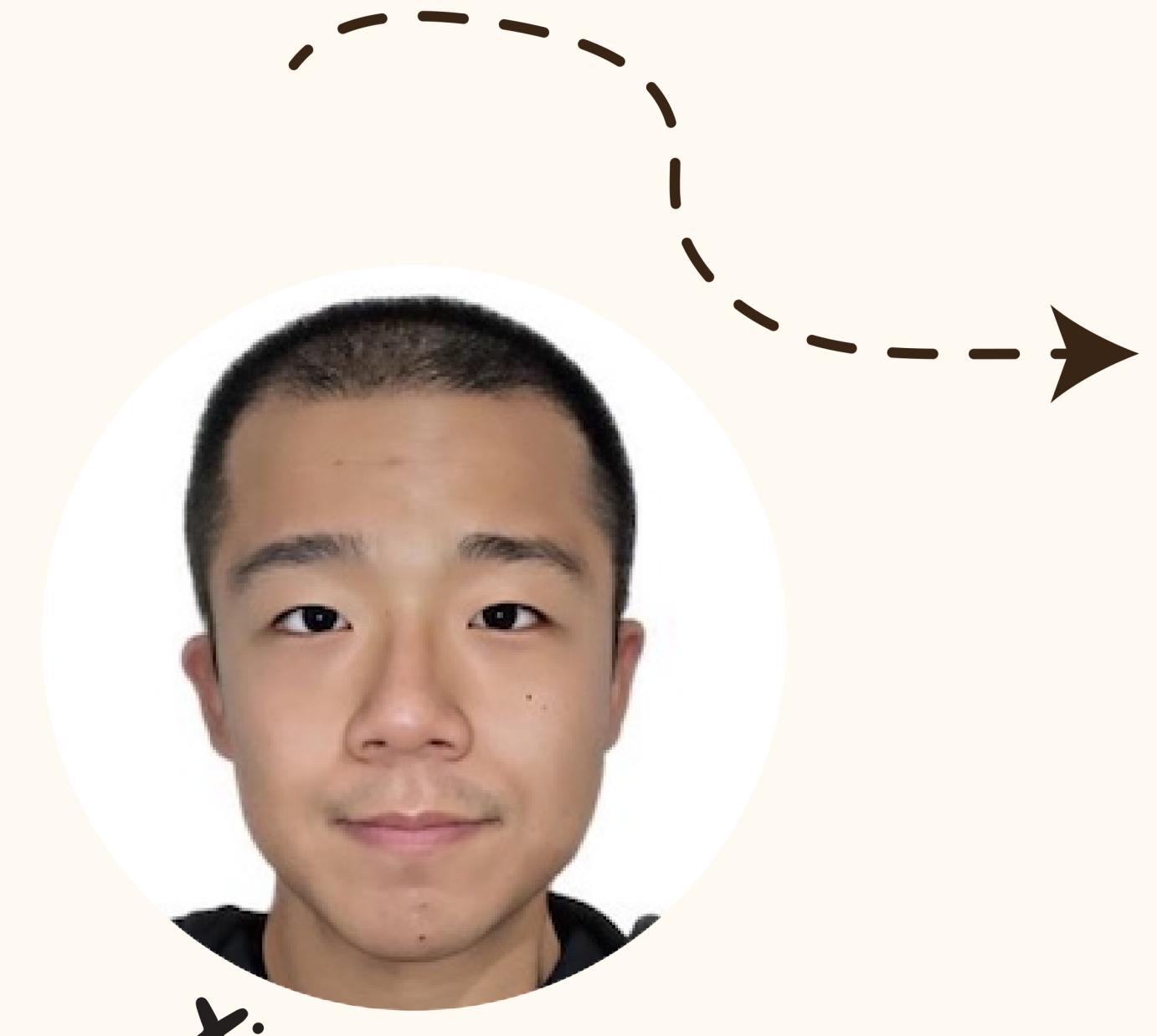


LET'S TRAVEL WITH MYWAY

Presented by: Xinzhe Yu & Yuka Toshima



Yuka Toshima



Xinzhe Yu

ABOUT US

Backend



API:test

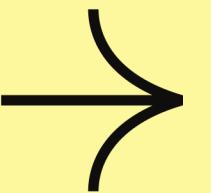


Frontend



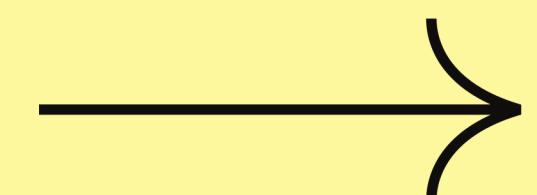
Redux

Frontend:test



Deployment

Challenge



Developing Environment

Root path

```
> backend  
> coverage  
> docs  
> frontend  
> node_modules  
  .babelrc  
  .env  
  .gitignore  
  package.json  
  README.md  
  yarn.lock
```

Backend

```
  < backend  
    < config  
    < controller  
    < coverage  
    < data  
    < middleware  
    < models  
    < node_modules  
    < routes  
    < test  
    < upload  
    < utils  
    .babelrc  
    .env  
    .gitignore  
    app.js  
    package.json  
    Procfile  
    README.md  
    seeder.js  
    server.js  
    yarn.lock  
  < controller  
    blogController.js  
    siteController.js  
    userController.js  
  < models  
    blogModel.js  
    siteModel.js  
    userModel.js  
  < routes  
    blogRoutes.js  
    siteRoutes.js  
    uploadRoutes.js  
    userRoutes.js
```

Frontend

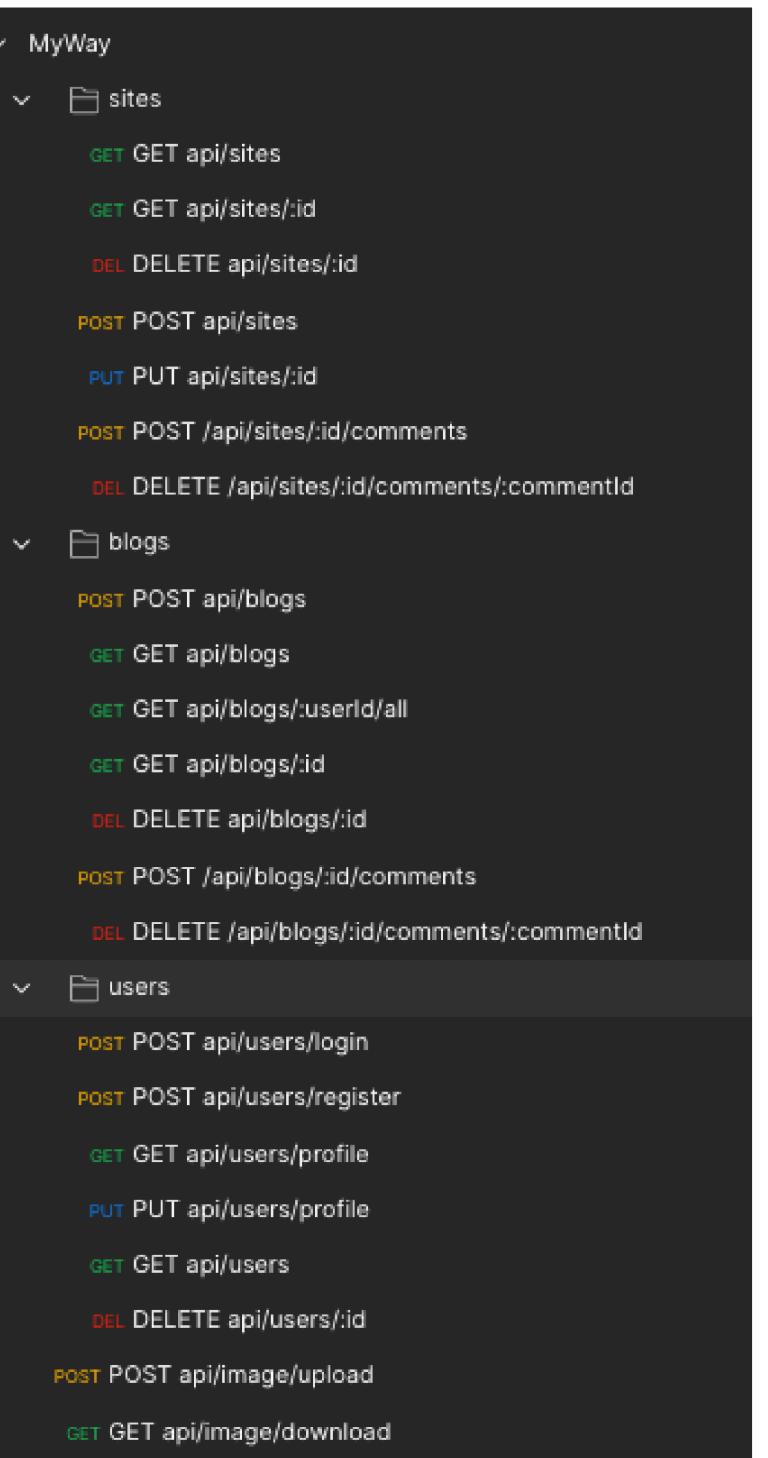
```
  < frontend  
    < coverage  
    < cypress  
    < node_modules  
    < public  
    < src  
    .gitignore  
    cypress.config.js  
    package.json  
    README.md  
    yarn.lock
```

Backend

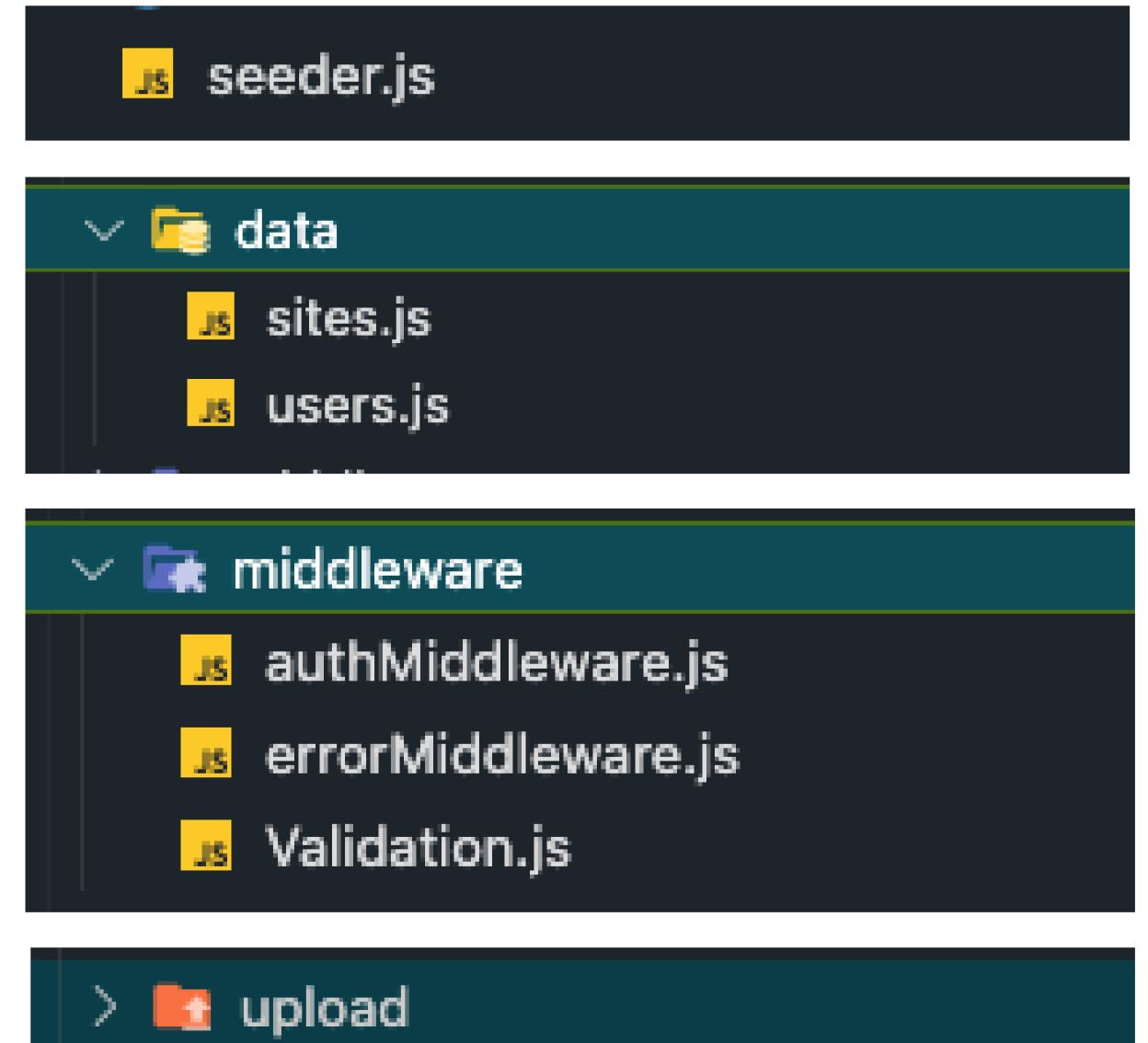
Packages

```
"dependencies": {  
    "bcryptjs": "^2.4.3",  
    "colors": "^1.4.0",  
    "cors": "^2.8.5",  
    "dotenv": "^16.0.3",  
    "express": "^4.18.2",  
    "express-async-handler": "^1.2.0",  
    "express-validator": "^6.14.2",  
    "helmet": "^6.0.0",  
    "jest-environment-jsdom": "^29.3.1",  
    "jsonwebtoken": "^8.5.1",  
    "mongodb-memory-server": "^8.9.5",  
    "mongoose": "^6.7.0",  
    "multer": "^1.4.5-lts.1",  
    "nodemon": "^2.0.20",  
    "supertest": "^6.3.1"  
},  
"devDependencies": {  
    "@babel/plugin-transform-modules-commonjs": "^7.19.6",  
    "jest": "^29.2.2"  
},
```

Postman



Others



Backend:test

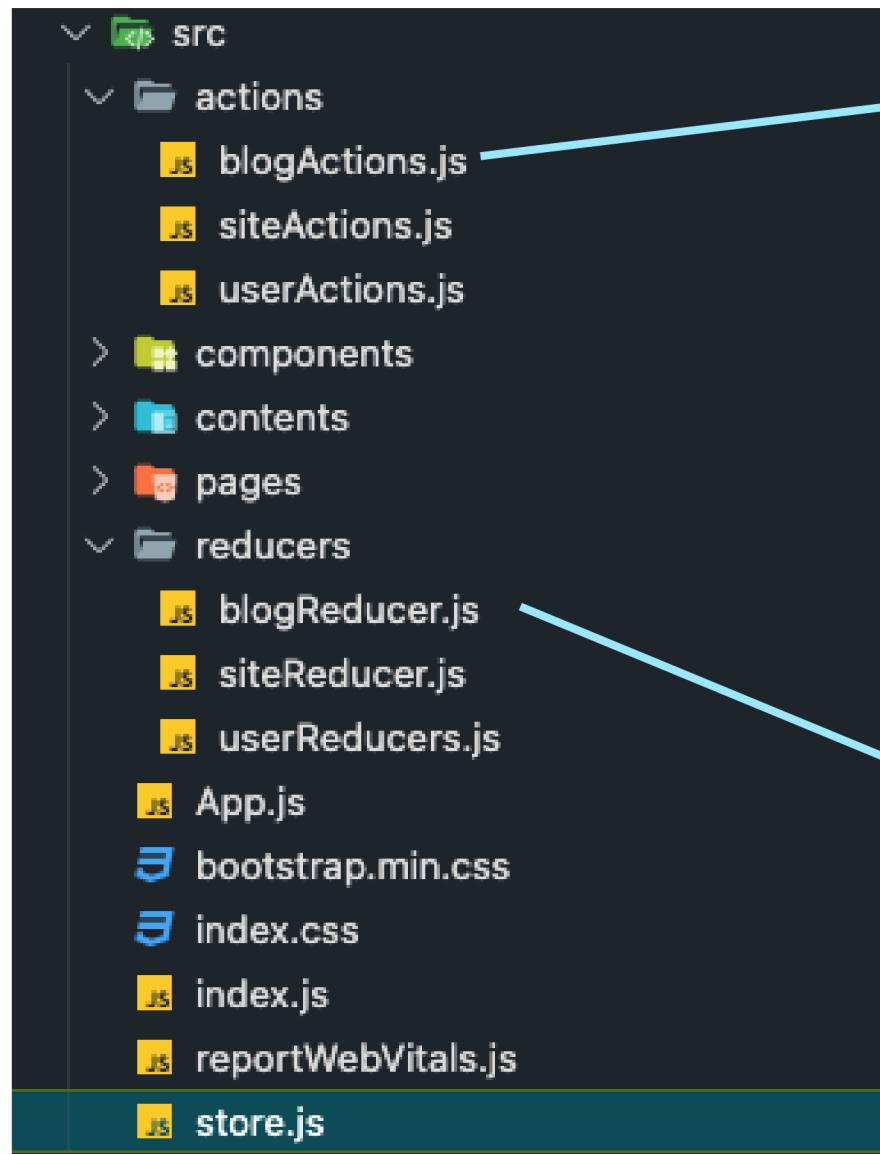
```
// API test: PUT /api/sites/:id
describe('PUT /api/sites/:id', () => {
  it("should post a new site with accurate data", async () => {
    const response = await request(app).post('/api/users/login').send({
      email: "admin@example.com",
      password: "123456"
    });
    const token = await response.body.token;
    const targetSite = await Site.findOne({name: "name of the site"})
    const targetSiteId = targetSite._id
    const site = await request(app).put(`/api/sites/${targetSiteId}`).send({
      name: "change name",
      description: "The Big Banana is Australia's ",
      rating: 4.0,
      numComments: 5,
      image: "/images/bigbanana.jpg",
      category: "park",
      "lat": 123,
      "lng": 11
    }).set('Content-Type', 'application/json').set('Authorization', `Bearer ${token}`)
    expect(site.body.name).toBe("change name");
  })
})
```

File	% Stmt	% Branch	% Funcs	% Lines	Uncovered Line #s
All files	66.03	52.5	66.66	66.53	
backend	90.9	100	50	90.9	17-18
app.js	90.9	100	50	90.9	
backend/controller	56.97	48.33	66.66	57.64	
blogController.js	51.78	54.54	62.5	51.78	13-14,25-26,50-51,70-71,79-92,100-112
siteController.js	50.72	18.75	50	52.23	23-24,76,89-90,100-113,121-131,139-149
userController.js	72.34	63.63	100	72.34	21-22,33-34,47-48,67-68,82,93-94,118-119
backend/middleware	66.66	68.75	80	66.66	
Validation.js	80	50	100	80	7
authMiddleware.js	64.7	70	100	64.7	21-22,26-27,36-37
errorMiddleware.js	62.5	75	50	62.5	3-5
backend/models	100	100	100	100	
blogModel.js	100	100	100	100	
siteModel.js	100	100	100	100	
userModel.js	100	100	100	100	
backend/routes	83.33	0	0	83.33	
blogRoutes.js	100	100	100	100	
siteRoutes.js	100	100	100	100	
uploadRoutes.js	42.85	0	0	42.85	17-19,23
userRoutes.js	100	100	100	100	
backend/utils	100	100	100	100	
generateToken.js	100	100	100	100	

Test Suites: 1 passed, 1 total
Tests: 19 passed, 19 total
Snapshots: 0 total
Time: 7.217 s, estimated 8 s
Ran all test suites matching /test\app.test.js/i.

Frontend

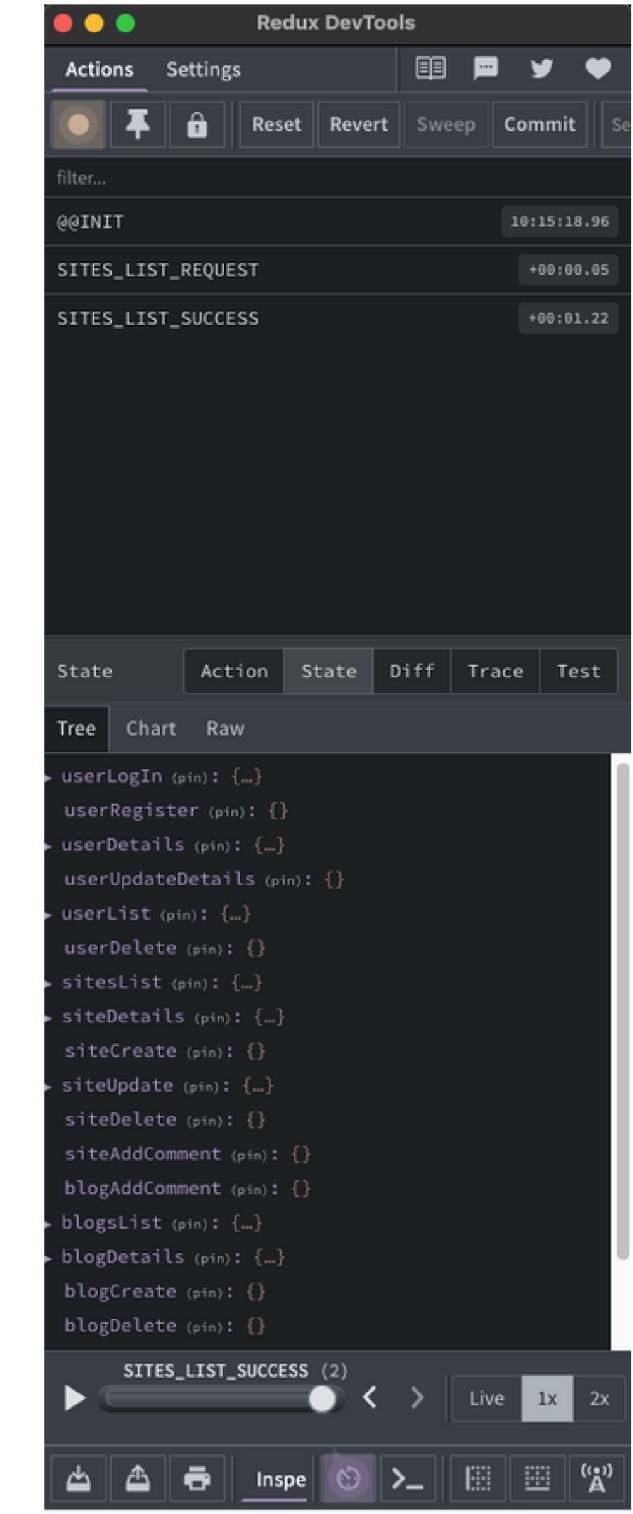
Redux



```
// delete comment to one blog
export const deleteCommentToOneBlog =
  (id, commentId) => async (dispatch, getState) => {
  try {
    dispatch({ type: DELETE_COMMENT_REQUEST });
    const {
      userLogIn: { userInfo },
    } = getState();
    const config = {
      headers: {
        Authorization: `Bearer ${userInfo.token}`,
      },
    };
    await axios.delete(
      `https://myway-backend.herokuapp.com/api/blogs/${id}/comments/${commentId}`,
      config
    );
    dispatch({ type: DELETE_COMMENT_SUCCESS });
  } catch (error) {
    dispatch({
      type: DELETE_COMMENT_FAIL,
      payload:
        error.response && error.response.data.message
        ? error.response.data.message
        : error.message,
    });
  }
};

// reducer to DELETE comment for a site
export const commentBlogDeleteReducer = (state = {}, action) => {
  switch (action.type) {
    case DELETE_COMMENT_REQUEST:
      return {};
    case DELETE_COMMENT_SUCCESS:
      return { success: true };
    case DELETE_COMMENT_FAIL:
      return { error: action.payload };
    default:
      return state;
  }
};
```

```
const reducer = combineReducers({
  // login, signup
  userLogIn: userLogInReducer,
  userRegister: userRegisterReducer,
  // MY HOME
  userDetails: getUserProfileReducer,
  userUpdateDetails: updateUserProfileReducer,
  // Admin user list page
  userList: userListReducer,
  userDelete: deleteUserReducer,
  // Find your way page, Admin Site list page
  sitesList: sitesListReducer,
  // Site detail page
  siteDetails: siteDetailReducer,
  // Admin Site list page
  siteCreate: siteCreateReducer,
  siteUpdate: siteUpdateReducer,
  siteDelete: siteDeleteReducer,
  // Site detail page
  siteAddComment: commentSiteAddReducer,
  // blog article page
  blogAddComment: commentBlogAddReducer,
  //blog page
  blogsList: blogsListReducer,
  // blog article page, blog Form
  blogDetails: blogDetailReducer,
  // MY HOME page
  blogCreate: blogCreateReducer,
  blogDelete: blogDeleteReducer,
  userBlogs: userBlogsReducer,
  blogCommentDelete: commentBlogDeleteReducer,
  siteCommentDelete: commentSiteDeleteReducer,
});
```



Frontend: test

Cypress

The screenshot displays the Cypress interface on the left and a browser preview on the right.

Cypress UI (Left):

- Specs:** Shows 2 green checkmarks and 0 red Xs.
- File Tree:** Shows the file `login.cy.js`.
- Test List:** Under `<Login /> page`, there are two tests:
 - ✓ should not log in if the user isn't exist
 - ✓ should log in when the user type the correct email and password and exist

Browser Preview (Right):

- Address Bar:** `http://localhost:3000/`
- Browser Headers:** Chrome 107, 1000x660
- Page Content:** A landing page for "MyWay" with navigation links: MyWay, ABOUT, BLOG, FIND YOUR WAY. A user "xinzhe" is logged in. A search bar says "Enter the city for checking the weather".

Surfing in Gold Coast

Deployment

Heroku

<https://github.com/Final-Project-CoderAcademy/Final-project/tree/main/backend>

1. Only deploy backend folder
2. Secure Node.js Applications with a Content Security Policy
3. Set cors, allow Netlify url of this app use the backend APIs.

Netlify

<https://github.com/Todd0554/myway-frontend>

1. Only deploy frontend folder
2. Cancel the proxy
3. Change the api url for each request in frontend

```
const { data } = await axios.post(  
  "/api/users/register",  
  { name, email, password },  
  config  
);
```

```
const { data } = await axios.post(  
  "https://myway-backend.herokuapp.com/api/users/register",  
  { name, email, password },  
  config  
);
```

Challenge

1. Upload image, download image with multer and show the downloaded image in frontend by using blob.
2. Use redux in frontend.
3. Babel settings.
4. Secure Node.js Applications with a Content Security Policy.