Index

A

addComment() function, 195
add.image() method, 273
addMarker() function, 193, 194
add.sprite() method, 264
allMarkdownRemark, 175
Amazon AWS, 203, 222
anims.create() method, 262
anims.generateFrameNumbers() method, 262
app.createNavBar() method, 28, 29
app.firstMouseEventDone property, 33
app.js files, 23, 24
Application programming interfaces (APIs), 7
app.mouseMoveHandler() method, 31, 32, 35
app.resizeHandler(), 31, 33, 35
app.slideIn() method, 33, 34
app.slideVal property, 35
app.start() function, 24, 27, 31
Async/await, 219
attach() method, 110
Axios, 217–221

B

Babel, 130–134
Backdrop component, 226
BitKeeper, 48
Branches, 47
Button component, 160

C

Cascading Style Sheets (CSS), 3
client.query(), 233
Collection index, 210
collection() method, 233
Command-line interface (CLI) tool, 63
componentDidMount() method, 148, 181, 215, 217
componentDidUpdate() method, 148
Components, React, 112–119
contentInner class, 26, 43
Content Management System (CMS), 10
Continuous deployment model, 8
create() method, 261
createPage() function, 174
Cross-origin request sharing (CORS), 296
CSS Grid, 40–42
currentMarkerText variable, 187, 192, 237

D, E

defaultProps, 142, 143, 146
descriptionValidator() function, 146
destroy() method, 270
Developer experience, 12
Dialog component, 164
DialogActions component, 166

© Frank Zammetti 2020
F. Zammetti, *Practical JAMstack*, https://doi.org/10.1007/978-1-4842-6177-4

DialogTitle component, 166
dialogVisible property, 162, 163, 167
document.createRange(), 191
documentDiv style, 169
document.getElementById(), 36
Document Object Model (DOM), 109
  JavaScript API, 109
  JavaScript API methods, 111
  virtual, 110
documentTemplate.js, 236
Dynamic content, 6

F

Facebook, 105
familyContainer style class, 37
FaunaDB, 199
  authentication, 228
  collection, creation, 209
  collection index, 210
  dashboard page, 208
  GraphQL queries, 207
  key requirements, 227
  NoSQL database, 206
  package.json, 210, 230
FaxJs, 106
fetch(), 17
File Transfer Protocol (FTP), 3
filter() method, 17
firstName property, 210
forEach() function, 174
Form submission handling, 202

G

gameCore.js file
  basic structure, 252
  create() method
    astronaut, 261–263
    fire, 264, 265
    keyboard input, 266
    sounds, 266
    spatial rupture, 267, 268
    stone track, 266
  createTrack() function
    coding, 270–273
    level data, 268–270
  init() function, 257
  load.spritesheet() function, 260, 261
  module-level variables, 253–256
  preload() method, 257–259
  update() method, 280, 281, 283–286
  startLevel() function, 277–279
gameOver() function, 281, 286–289
gameState.gameRunning, 289
gameState.trackMoveCount, 281
Gatsby
  Babel, 134
  build process, 70
  code
    configuration file, 73–75
    header.js, 83–85
    image.js, 85–87
    index.js, 75, 77–79
    404.js file, 92
    layout.js file, 79–83
    page-2.js, 90, 91
    seo.js, 87–90
    typescript.tsx, 91, 92
  command-line tool, 61
  generated website, 64
  GraphQL (*see* GraphQL)
  installation, 62–64
  plugins, 61, 67, 68
  project structure, 65–67
  react websites, 61

starter, 67
themes, 69
Webpack, 70–72
Webpack configuration, 274
gatsby build, 211, 214, 216
ggatsby-config.json file, 82, 156
gatsby develop, 216, 296
gatsby-node.js file, 274
gatsby-plugin-manifest plugin, 296
gatsby-plugin-offline, 296
gatsby-transformer-remark plugin, 170
Get() method, 233
getAllMarkerKeysFromStorage() function, 189, 221–223
getAllMarkerKeysFromStorage.js, 232–234
getDerivedStateFromProps() method, 147, 148
getMarkerFromStorage() method, 223, 237
getSnapshotBeforeUpdate(), 148
Git flow, 47
GitHub, 53, 54
GitHub pages, 56, 57
Git repo, 56, 206
global.showComments(), 213
GraphQL, 61, 85
arguments and filtering, 96–99
execute queries, 94
explorer, 95
getCrewmembers, 94
limiting/skipping, 100
query, 93–95
query operation type, 94
server, 101
sorting, 101
types, queries, 102, 103
GraphQL query, 158, 250
Grid component, 185
Material-UI Grid component, 184, 188

H

handleDialogClose() function, 182, 195
handleDialogSave() function, 166, 182, 195, 196
Header component, 80, 83
Hello-world starter
index.js file, 246–250
JSON-level data, 250, 251
Helmet component, 87
hrStyle object, 248, 249
HTML5 games, 244

I

Image component, 778, 787
images.astronaut.anims.resume(), 263
index.html file, 22, 24–36
IndexPage component, 161, 166, 215, 216
IndexPage React component, 161, 162
inEvent object, 33
init() function, 247
innerContent style class, 36

J, K

JAMboard app
document list page, 152
Gatsby, 155
home/document list page
coding, 169–171
gatsby-node.js, 172–174
IndexPage component, 158
IndexPage React component, 161, 162
markdown support, 160, 161
Material-UI Toolkit, 159, 160

JAMboard app (*cont.*)
prompting for username, 163, 165–167
querying for data, 174, 175
rendering, component, 162, 163
site metadata, 167, 168
home page, 151
trimmed down
gatsby-config.json file, 156
index.js, 157
404.js page, 157
Whiteboard Page (*see* Whiteboard Page)
JAMJumper, 243, 246
JAMstack
advanced app, 15–17
advantages, 11–13
build, website, 10, 11
continuous deployment model, 8
description, 5, 6
disadvantages, 13, 14
dynamic content, 6
history, 2–5
HTML, CSS, and JavaScript, 6
misconceptions, 14, 15
oriented services, 8
pillars, 6
website, 6, 7, 15
workflow, 8, 9
JavaScript, 3, 7, 15
JavaScript DOM API methods, 109, 111
JavaScript Object Notation (JSON), 74
JavaScript XML (JSX), 128
json() method, 17
JSON.parse(), 252
JSX, 136–140
jumpData.isJumping, 282, 284
jumpData.jumpTicks, 284
jumpData.phase, 283

L

Lambda functions, 203, 222
Layout component, 77–80
Library/framework, 4
Link component, 78, 79
linkifyMarker() function, 189
Linux, Apache, MySQL, PHP (LAMP) stack, 6
ListItem components, 188
ListItemText component, 188
loadMarkers() function, 181, 188
Local storage, 215
Long-term support (LTS), 62

M

Manifest file, 290, 291
map() method, 169
MarkdownRemark(), 197
markdownRemark object, 185
maskScreen() function, 225, 226
Match() function, 231
MaterialButton, 140
MaterialButton class, 135
MaterialButton component, 129, 140
Material-UI, 161, 225
Material-UI library, 160
Material-UI toolkit, 159, 160
maxWidth prop, 164
metaDescription variable, 89
MongoDB, Express, Angular, Node (MEAN) stack, 6
mousemove events, 34

myFamily.html file, 36–42
myWork style class, 30

N

Netlify
  account dashboard, 204
  Amazon AWS, 203
  Amazon's AWS, 203
  common services, 202
  continuous integration/continuous deployment, 202
  deploy, site, 205
  functions, 206, 226, 227
  GitHub account, 203, 204
  history, 202
  resource optimization functions, 202
  static sites, 201
  web hosting company, 202
netlify.toml file, 226, 228
node.frontmatter property, 170
node_modules directory, 132
Node Package Manager (NPM), 62
NoSQL database, 207
npx command, 132

O

onMouseOut event handler, 171

P

Paginate() function, 231
Paper component, 184, 185
particles.createEmitter() method, 268
Phaser, 244, 274
PHP stack, 106
physics.add.sprite() method, 263
placeholderImage property, 86
Plain Old JavaScript Objects (POJOs), 110
Plugins, Gatsby, 67, 68
Polyfill, 74, 131
preload() method, 257–259, 261, 266
Progressive Web App
  description, 289
  history and perspective, 293, 294
  HTTPS, 293
  JAMJumper, 290, 295, 296
  manifest file, 290, 291
  Netlify hosting, 296–298
  Service Worker, 291–293
  vs. standard webapp, 292
Prop expressions, 140
Props, react, 119, 120
propTypes, 80, 143, 145, 146
Pushing, 49

Q

query() function, 231
query.Collection() method, 235

R

React, 61, 76
  Babel, 130–134
  compile JSX, 134
  components, 112–119
  critical points, 127
  DOM methods, 111
  FaxJs, 106
  history, 105–107
  interface, 108
  JSX, 128, 129, 136–140
  MaterialButton component, 129, 130
  pillars, 109

React (*cont.*)
  props, 119, 120
  state, 121, 122, 124
  style, 124–126
React components, 78, 102
  class, 118, 122
  default props, 141–143
  library, 159
  lifecycle events, 147, 148
  typing props, 143–146
React.createElement(), 115, 116, 119, 122, 125, 129, 138, 140
react-dom package, 112
ReactDOM.render(), 116, 138
RealMedia video streams, 2
Ref() method, 233
Remote repo, 50–52
removeAllRanges() function, 192
render() method, 78, 81–83, 88, 103, 118, 126, 146, 148, 163
Representational state transfer (REST), 93
requestAnimationFrame(), 254
restartGame argument, 278
RESTful interface, 4
Robust toolkits, 108

S

saveMarkerToStorage() function, 196, 224
saveMarkerToStorage.js, 234, 235
Scalability, 11
Search engine optimization (SEO), 1, 77
Security, 12
SEO component, 87–90
Server-Side Rendering, 211
Service Worker, 291–293
setState() method, 122, 123
setTimeout(), 254
shouldComponentUpdate(), 148
showComments() function, 181, 191–193, 213
Sift library, 96
Single-page apps (SPAs), 105
siteMetadata branch, 74
Source Code Management (SCM), 4
  basic operations
    checking repo status, 52
    commit message, 52
    local Git repo, 50
    pulling, 53
    remote repo, 52
    staging and committing, 51
  BitKeeper, 48
  branches, notion of, 47
  conflicts, 46
  distributed, 49
  Git, 48, 49
  GitHub, 53, 54
  repository, 47
spinnerVisible property, 225
spinnerVisible state object property, 226
Splash page, 20
spritesheets, 259, 260
start() function, 23, 116
startLevel() function, 250, 277–279, 287
startsWith(), 228, 232
State, react, 121, 122, 124
Static site generator, 211
Static HTML page, 6
Static site generator, 59, 60
Static web, 5
Steep learning curve, 14
StorageManager module, 203
StorageManager, remote storage
  Axios, 217–221
  coding

getAllMarkerKeysFromStorage()
function, 221, 222
getMarkerFromStorage()
method, 223
saveMarkerToStorage()
function, 224
Style, react, 124–126
styles.css file, 22–24
surroundContents() method, 192

T

Template component, 225
TextField component, 196
textNodes array, 191
then() function, 233
this.props, 119
tickCount, 286
trackMoveCount attribute, 255
tweens.add() method, 265
typescript.tsx, 91, 92

U

update() method, 280, 281, 283–286
useStaticQuery and GraphQL, 80, 85

V

Vanity website
coding
contactMe.html file, 45
index.html, 24–36
myFamily.html, 36–42
myHobbies.html file, 44, 45
myWork.html file, 43
styles.css and app.js, 23, 24
directory structure/constituent files, 22
GitHub pages, 56
home page, 20
hosting, 55
my work life page, 21
Virtual COM approach, 111
Virtual DOM, 110

W

Web-based game, 244
WebGL, 244
Webpack, 70–72, 274
Whiteboard Page
coding, 178, 179
addComment() function, 195
addMarker() function, 193, 194
component, building, 180, 181
component, rendering,
181, 183–186, 188
data querying, 197
handleDialogSave()
function, 195, 196
linkify marker, 189–192
loading markers, 188
showing marker
comments, 192, 193
StorageManager, 177
window.showComments() function, 191

X, Y, Z

XMLHttpRequest object, 217, 218