**Preface.....vi**

**Introduction .....8**

**Managing Software Complexity .....14**

EXPLAINING COMPLEXITY.....16
DETERMINE THE ROOT CAUSES OF COMPLEXITY.....18
A PHILOSOPHY OF SOFTWARE DESIGN.....19
OUT OF THE TAR PIT.....20
SIMPLE MADE EASY.....21
NO SILVER BULLET.....22
SYSTEM DESIGN AND THE COST OF ARCHITECTURAL COMPLEXITY .....24
HOW CAN TEAMS MANAGE COMPLEXITY?.....25
THE BEST SOLUTIONS ARE SIMPLE BUT NOT SIMPLISTIC. .....26
SOMETIMES, (ESSENTIAL) COMPLEXITY HAS TO LIVE SOMEWHERE.....28
WHAT ARE ONGOING CHALLENGES IN MANAGING COMPLEXITY?.....29
CONCLUSION.....29

**Modularity .....31**

MODULES IN JAVASCRIPT.....32
LAZY-LOADING.....41
CODE-SPLITTING .....46
WRAP UP .....51

**Performance.....52**

UNDERSTANDING HOW BROWSERS WORK.....53
UNDERSTANDING AND REDUCING THE COST OF JAVASCRIPT.....57
OPTIMIZE INTERACTIONS .....60
NETWORKING .....60
REDUCING THE IMPACT OF THIRD-PARTY DEPENDENCIES .....61
RENDERING PATTERNS.....63
OPTIMIZING PERCEIVED PERFORMANCE .....65
OPTIMIZING PERFORMANCE - THE HIGHLIGHTS.....69
PERFORMANCE CULTURE .....71

**Design Systems.....73**

CODING STYLE GUIDES.....74
DESIGN TOKENS .....77
COMPONENT LIBRARIES .....82
ACCESSIBILITY.....85

PERFORMANCE....86
DOCUMENTATION....87
CASE STUDIES....89
WRAP UP....93

**Data Fetching....95**

BROWSER APIS AND SIMPLE HTTP CLIENTS....96
MORE SOPHISTICATED DATA-FETCHING LIBRARIES....98
TIPS FOR EFFICIENT DATA-FETCHING....113

**State Management....119**

MANAGING DATA BETWEEN COMPONENTS....120
PROP DRILLING....123
SIMPLE STATE MANAGEMENT....126
DEDICATED STATE MANAGEMENT LIBRARIES....128
FINAL CONSIDERATIONS....133

**Internationalization....136**

UTILIZE THIRD-PARTY LOCALIZATION LIBRARIES....139
DYNAMIC LOADING....145
HANDLING PLURALS ACROSS LANGUAGES....147
FORMAT DATES, TIMES, AND NUMBERS....149
CONSIDER RIGHT-TO-LEFT (RTL) LANGUAGES....153
WRAP UP....160

**Organizing Code....161**

FOLDER AND FILE STRUCTURE....162
NAMING CONVENTIONS....166
BARREL EXPORTS....167
OTHER GOOD PRACTICES....168
WRAP UP....173

**Personalization and A/B Testing....174**

PERSONALIZATION....175
A/B TESTING....178
FEATURE FLAGS....185
WRAP UP....189

**Scalable Web Architecture....190**

SCALABILITY....191

CHARACTERISTICS OF A SCALABLE APPLICATION....196
WHERE DO KUBERNETES AND DOCKER FIT IN?....198
WHERE DO COMPANIES LIKE VERCEL AND NETLIFY FIT IN?....201
WRAP UP....202

**Testing....204**

UNIT TESTS....205
END-TO-END TESTS....211
INTEGRATION TESTS....220
SNAPSHOT TESTS....223
HOW SHOULD WE TEST OUR APP?....227

**Tooling....233**

VERSION CONTROL (GIT)....234
CONTINUOUS INTEGRATION....236
BUNDLERS....238
LINTING....239
LOGGING AND PERFORMANCE MONITORING....241
WRAP UP....243

**Technical Migrations....244**

DIFFERENT MIGRATION STRATEGIES....245
MIGRATION STRATEGY....247
CODEMODS....249
THE ROLE OF GENERATIVE AI....254

**TypeScript....259**

TYPE SAFETY....260
BUILD TOOLS & TYPESCRIPT....262
CONFIGURATION & LINTING....263
REACT + TYPESCRIPT....266
DECLARATION FILES....286
AUTO-GENERATING TYPES FROM AN API....288
MIGRATING AN EXISTING REACT APP TO TYPESCRIPT....295

**Conclusions....300**