### **About the reviewer**

**Akash Ranjan** is a Python professional, with 6+ years of industry experience. He has developed, deployed, and managed large-scale enterprise applications. He has extensive experience in building APIs and designing microservice-based application architecture.

# **Table of Contents**

**Preface** 

Part 1: An Introduc	tio	n to FastAPI		
1				
<b>Getting Started with FastA</b>	Getting Started with FastAPI			
Technical Requirement Git basics Installing Git Git operations Git branches	4 4 5 8	Package management with pip Installing pip Setting up Docker Dockerfile	11 12 13	
Creating isolated development environments with Virtualenv Creating a virtual environment Activating and deactivating the virtual environment	<b>9</b> 9	Building a simple FastAPI application Summary	1! 16	
2 Routing in FastAPI				
Technical requirements Understanding routing in FastAPI	18 18	Validating request bodies using Pydantic models  Nested models	<b>2</b> 3	
Routing example  Routing with the  APIRouter class	18 <b>19</b>	Path and query parameters Path parameters Query parameters	27 27 30	

viii	Tabl	e of	Conte	nts
VIII	Tabi	e or	Conte	11

Request body FastAPI Automatic Docs	<b>30</b> 31	Building a simple CRUD app Summary	37 42
3			
Response Models and Erro	r Ha	ndling	
Technical requirements	44	Status codes	44
Understanding responses in FastAPI What is a response header? What is a response body?	<b>44</b> 44	Building response models Error handling Summary	45 48 52
4 Templating in FastAPI			
Technical requirements Understanding Jinja	53 54	Macros Template inheritance	57 58
Filters Using if statements Loops	54 56 56	Using Jinja templates in FastAPI Summary	58 67
Part 2: Building and Applications	l Se	curing FastAPI	
5 Structuring FastAPI Applica	atioı	ns	
Technical requirements Structuring in FastAPI	72	Implementing the models Implementing routes	74 78
applications Building an event planner application	<b>72</b> 73	Summary	87

_

## Connecting to a Database

Technical requirements Setting up SQLModel	90	Document	104
	90	Initializing the database	106
Tables	90	CRUD operations Create Read	<b>110</b>
Rows	91		110
Sessions	92		110
Creating a database Creating events Read events Update events	<b>92</b> 97 99 101	Update Delete routes/events.py routes/users.py	111 112 112 114
Delete event  Setting up MongoDB	102 <b>104</b>	Summary	120

#### 7

## **Securing FastAPI Applications**

Technical requirements	122	Handling user authentication	132
Authentication methods in FastAPI	122	Updating the application	133
Dependency injection Creating and using a dependency	123 123	Updating the user sign-in route Updating event routes Updating the event document	133 136
Securing the application		class and routes	140
with OAuth2 and JWT	124	Configuring CORS	145
Hashing passwords Creating and verifying access tokens	125 129	Summary	146