

# Technology Stacks MVP

**MVP:** Building an identical CRUD task-app in each one of my current stacks. It will build understanding in these technologies and serve as a blueprint for more complex processes.

## Stacks:

- 1) Flask - React
- 2) Express - React
- 3) Spring Boot - React
- 4) NestJS - React
- 5) NextJS
- 6) Express - Angular
- 7) Django
- 8) HTML, CSS and JavaScript

## App Plan:

### *Backend:*

- Application written in one page of code
- URL endpoint to test access
- A value passed to the frontend statically (w/o database access)
- A create endpoint
- A read endpoint
- An update endpoint
- A delete endpoint

### *Frontend:*

- One page with static backend text and button to redirect to form
- A form page with a CRUD form
- Dynamic display of todo items
- Buttons to create, update and delete

### *Database:*

- Filled with one value automatically
- Extremely simple design

Ex.

```
Table todos {  
    primary_key: number,  
    text: string  
}
```

**Notes:**

- Identical App
- No Styling
- Have frontends / backends and databases self-contained and automatically interchangeable

**Additional Technology:**

- Docker
- AWS Services
- PostgreSQL
- Redis
- Heroku

**Specifics:**

- Serverless Technology
- AWS Elastic Beanstalk
- AWS S3
- AWS Lambda
- AWS ECR
- AWS ECS
- AWS EC2

**API Design:**

- **GET /helloworld**
- **GET /**
- **GET /todo**
- **POST /todo**
- **PUT /todo/:id**
- **DELETE /todo/:id**