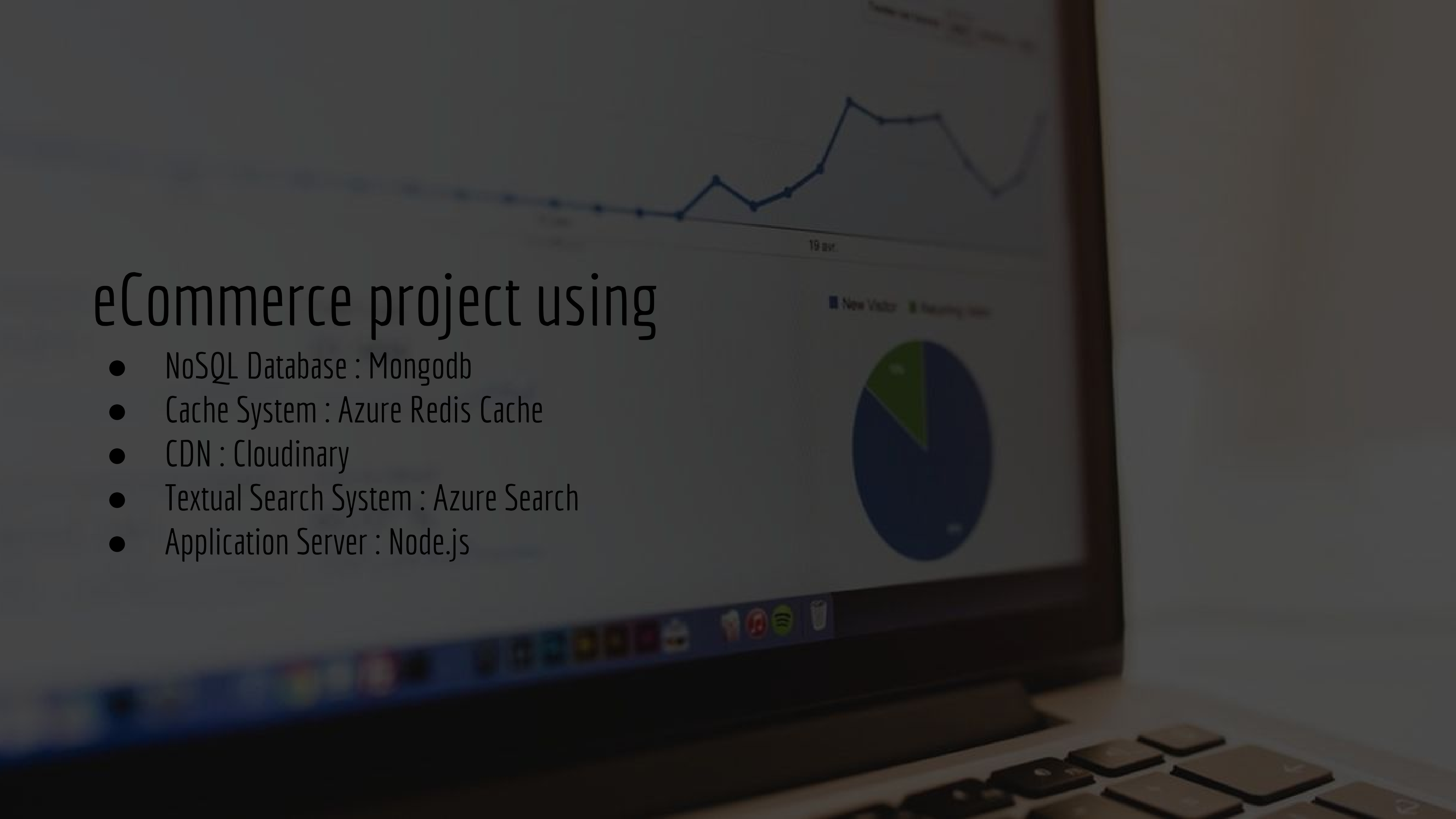
BIG DATA PROJECT

Submitted by-
Diksha (U33189261)
Jijo (U12605351)
Pratul (U56790213)
Pradeep (U51084028)
SRK Abhimanyu (U61023879)

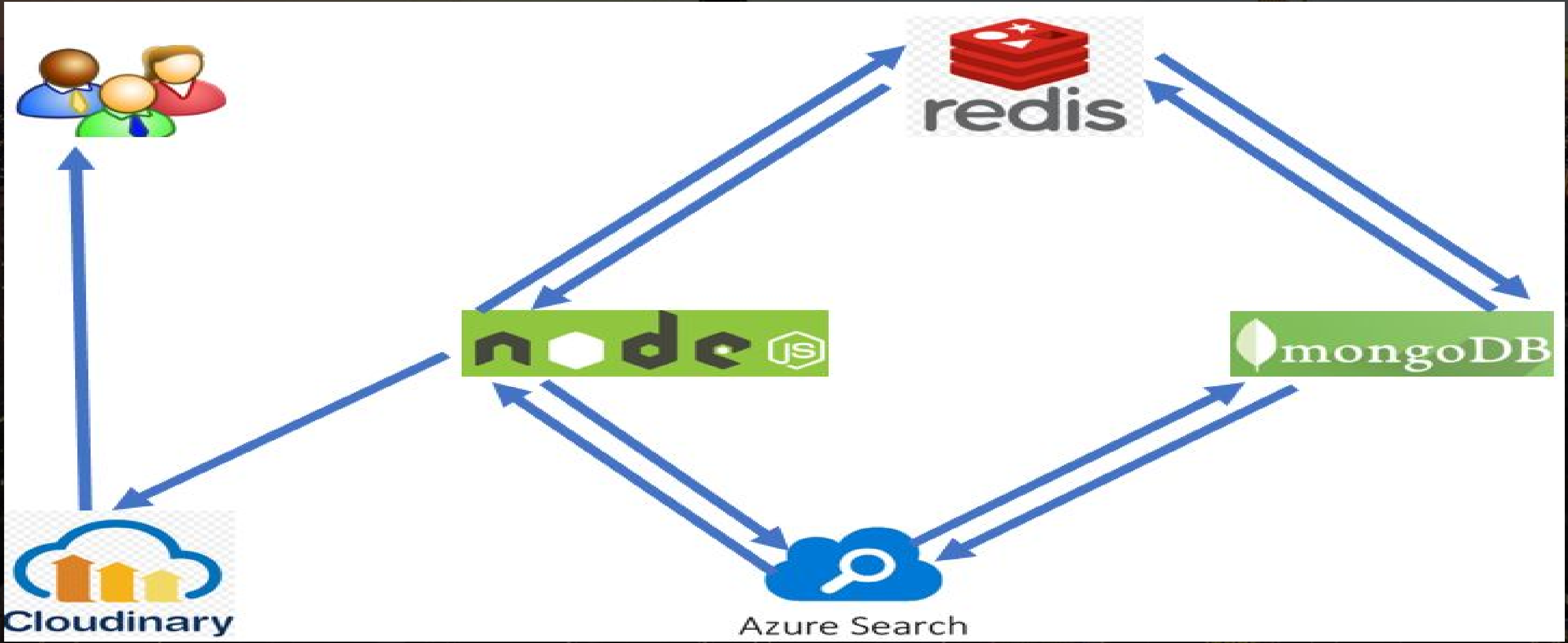
Submitted to :
Prof Dutta



eCommerce project using

- NoSQL Database : Mongodb
- Cache System : Azure Redis Cache
- CDN : Cloudinary
- Textual Search System : Azure Search
- Application Server : Node.js

The Architecture

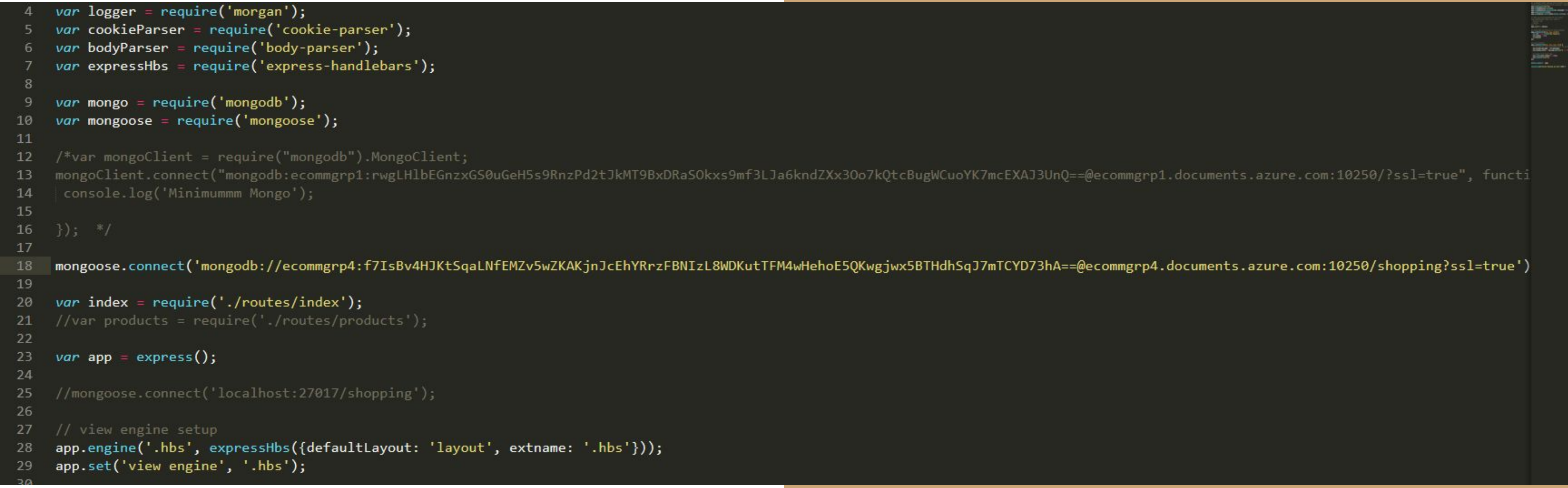


Mongodb on AZURE (NoSql DocumentDB)

- Use RoboMONGO for Data Loading
- Configured Azure Documentdb on cloud
- Validated Connection profile and SSL gateway
- Ideal for pay-as-you-go services and monitoring performance

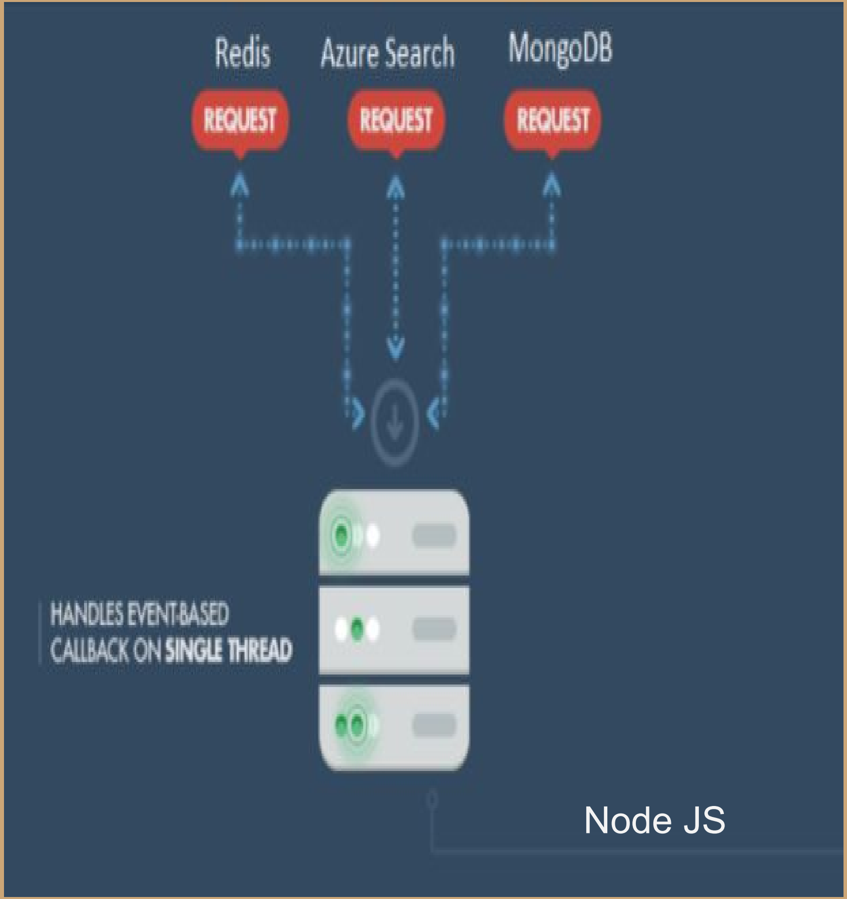
Why did we use Mongodb?

Scaling on demand
Flexible Schema
Geo Spatial Queries
Real Time analytics



```
var logger = require('morgan');
var cookieParser = require('cookie-parser');
var bodyParser = require('body-parser');
var expressHbs = require('express-handlebars');

var mongo = require('mongodb');
var mongoose = require('mongoose');

/*var mongoClient = require("mongodb").MongoClient;
mongoClient.connect("mongodb:ecommgrp1:rwgLHlbEGnzxGS0uGeH5s9RnzPd2tJkMT9BxDRaSOkxs9mf3LJa6kndZXx3Oo7kQtcBugWCuoYK7mcEXAJ3UnQ==@ecommgrp1.documents.azure.com:10250/?ssl=true", functi
 console.log('Minimummm Mongo');

});  */

mongoose.connect('mongodb://ecommgrp4:f7IsBv4HJKtSqaLNfEMZv5wZKAKjnJcEhYRrzFBNIzL8WDKutTFM4wHehoE5QKwgjwx5BTHdhSqJ7mTCYD73hA==@ecommgrp4.documents.azure.com:10250/shopping?ssl=true')

var index = require('./routes/index');
//var products = require('./routes/products');

var app = express();

//mongoose.connect('localhost:27017/shopping');

// view engine setup
app.engine('.hbs', expressHbs({defaultLayout: 'layout', extname: '.hbs'}));
app.set('view engine', '.hbs');
```

NodeJS Implementation



Node JS

NodeJS Packages

Front End

Express
Bootstrap
hbs

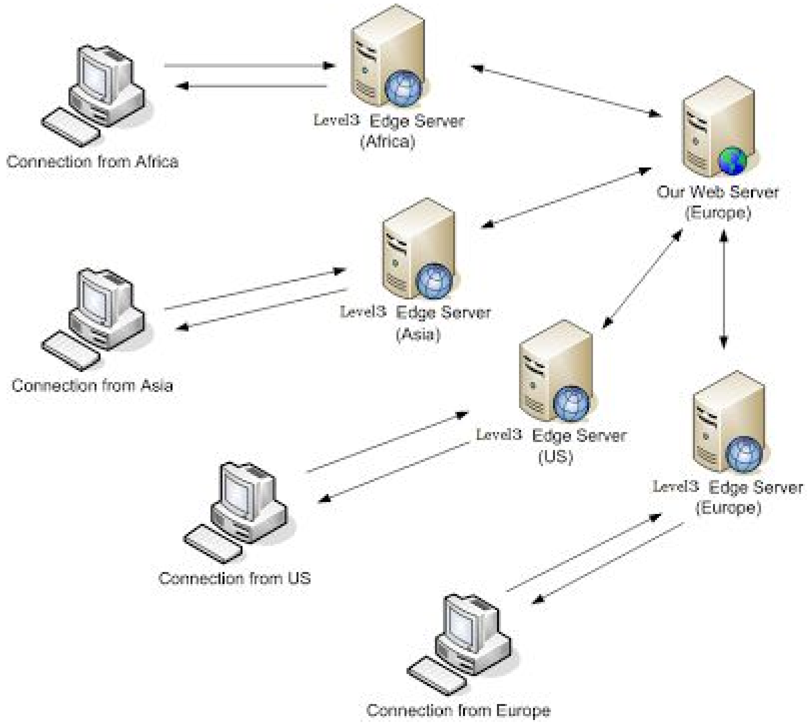
DB & SEARCH CONNECTION

Mongoose (Mongodb)
Redis
Node-azure-search

CDN

Stores static web content

Provides maximum throughput



AZURE SEARCH

- Cloud search-as-a-service solution by Microsoft.
- Uses Apache Lucene and Microsoft’s natural language processing technology.
- Built on Elasticsearch.
- Storage capacity is 50 MB with one replica and one partition.
- Used azure-search package in node.js to integrate azure search to our application.

Redis Cache

- Azure Redis Cache for our applications caching.
- It gives access to a secure dedicated Redis Cache by Microsoft and is accessible from any application within Azure.
- High throughput and consistent low-latency data.
- Standard tier Redis cache with 250 MB capacity.

Web scraping using python

Web scraping has been done using

- LXML
- Beautiful SOAP
- Python
- Amazon product advertising API

Web deployment - Windows Azure

Web deployment has been done using

- Visual Studio 2017
- Windows Azure app service
- Node.Js deployment packages for windows azure

Steps to achieve the target

Step 1

Built UI for the application on the local system

Step 2

On cloud we created services- Redis Cache, Azure Search , Mongodb

Step 3

We integrated the cloud services into NodeJS Server and finally deployed the application on Cloud

Future Scope

- Recommendation System
- Big data Analytics
- Big data Processing